



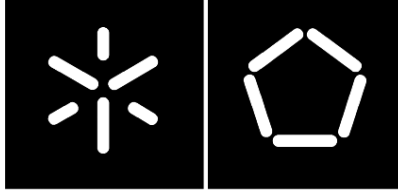
Universidade do Minho

Escola de Engenharia

Cristiano Gonçalves Pendão

Recolha de Dados de Movimento em Dispositivos Móveis Pessoais

Smartphone-based Human Motion Data Collection



Universidade do Minho
Escola de Engenharia

Cristiano Gonçalves Pendão

**Recolha de Dados de Movimento em
Dispositivos Móveis Pessoais**

**Smartphone-based Human Motion Data
Collection**

Tese de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efectuado sob a orientação de
Professor Doutor Adriano Moreira
Professora Doutora Maria João Nicolau

“If you’re going through hell, keep going.”

Winston Churchill

Agradecimentos

Neste momento, em que completo a etapa mais importante do meu percurso académico, meta fundamental para o meu crescimento pessoal e profissional, gostaria de expressar o meu reconhecimento a todos aqueles que contribuíram de forma direta e indireta para que este momento fosse possível.

Foi um privilégio realizar esta dissertação sob a orientação do Professor Doutor Adriano Moreira e da Professora Doutora Maria João Nicolau. Sem a sua ajuda e disponibilidade esta dissertação não seria possível. Os seus conhecimentos, conselhos e orientações foram de valor inestimável, pelos quais sou extremamente grato.

Ao meu pai António, à minha mãe Maria do Carmo e aos meus irmãos Fernando e Luísa, agradeço por me incentivaram e apoiarem inequivocamente, não só nesta etapa, mas durante todo o meu percurso académico. Também para eles a minha mera expressão de agradecimento não é suficiente.

À minha namorada Catarina, sou especialmente grato pelo apoio incondicional e pela paciência inigualável. Não existem palavras que expressem o quanto és importante.

A todos os meus amigos, agradeço pelos bons momentos, um especial e sincero obrigado ao Luís Gonçalves e ao Luís Nascimento, insubstituíveis companheiros de estudo e de grupo ao longo destes anos.

Resumo

Esta dissertação enquadra-se no Projeto SUM (Sensing and Understanding Human Motion Dynamics). O principal objetivo deste projeto é o estudo do movimento das pessoas num determinado meio, e o estabelecimento de modelos matemáticos para prever o seu movimento e a sua interação com os elementos presentes nos espaços físicos.

Os dispositivos móveis pessoais têm um forte impacto no dia a dia dos seus utilizadores, fazendo parte das suas rotinas diárias. Além de que, a maioria destes dispositivos estão equipados com diversos sensores e interfaces que podem ser utilizados para estudar a mobilidade humana.

O propósito desta dissertação centra-se no desenvolvimento de uma aplicação para dispositivos móveis, que permite a recolha de dados dos vários interfaces dos dispositivos (Wi-Fi, GPS, Bluetooth e células GSM) e o seu envio para um servidor, para posterior processamento e análise.

Para motivar a utilização da aplicação e maximizar o número de utilizadores, garantindo a utilização durante um longo período de tempo, é necessário criar um serviço atrativo integrado na aplicação e estabelecer um modelo que recompense e motive os utilizadores. Por outro lado, é essencial considerar as questões de privacidade, que este tipo de estratégias de recolha de dados levanta e considerar o seu impacto na utilização da aplicação, sendo necessário a integração de mecanismos que garantam a privacidade e segurança dos utilizadores.

Adicionalmente, é fundamental minimizar o consumo de recursos pela aplicação, como a utilização do processador e da memória, que nos dispositivos móveis são mais restritos do que nos restantes dispositivos, mas também reduzir os consumos energéticos de modo a maximizar a autonomia da bateria do dispositivo. Tudo isto pode ser alcançado através da criação de algoritmos eficientes para a recolha de dados, recorrendo às tecnologias do dispositivo, como por exemplo os diversos sensores, e utilizando comunicação oportunista para o envio dos dados recolhidos.

Abstract

This dissertation falls within the Project SUM (Sensing and Understanding Human Motion Dynamics). The main goal concerning this project is the study of people's movement in an enclosed space, and it tries to establish mathematical models to predict the movement of people and its interaction with the elements present in physical spaces.

Personal mobile devices have a heavily impact in the daily life of their users, and make part of their daily routines. In addition, most of these devices are equipped with several sensors and interfaces that may be used to study human mobility.

The purpose of this dissertation is to develop an application for mobile devices that allows the collection of data from several device's interfaces (WiFi, GPS, Bluetooth and GSM cells) and send the data to a server for later processing and analysis.

To motivate the use of the application and maximize the number of users, ensuring an use for a long period of time, it becomes necessary to create an appealing service integrated in that application, and also establish a model that rewards and motivates their users. On the other hand it is necessary to take into account the privacy issues brought up by this type of strategy in collecting data, and consider its impact in the use of the application, being necessary the integration of mechanisms for ensuring the user's privacy and security.

Moreover it is necessary to minimize resource consumption by the application, such as processor and memory resources, which in mobile devices are more restricted than in other devices and also, to maximize the device's autonomy, the energy resources need to be taken in account. All this can be achieved through the creation of efficient algorithms for data collection, using the device technology, such as the several available sensors, and using opportunistic communication to transmit the collected data.

Conteúdo

Agradecimentos	v
Resumo	vii
Abstract	ix
Lista de Figuras	xiv
Lista de Tabelas	xvii
1 Introdução	1
1 Enquadramento	1
2 Objetivos	2
3 Metodologia	3
4 Estrutura da Dissertação	4
2 Estado da arte	5
1 Objetivos do estudo do movimento humano	5
2 <i>Sensing</i> Colaborativo	7
2.1 <i>Sensing</i> Oportunista e Participativo	9
3 Dispositivos Móveis Pessoais como Sensores	10
3.1 Segurança, Privacidade e Integridade	11
4 Projetos de <i>Sensing</i>	13
5 Formas de Recompensa	18
5.1 Modelos de Negócio	19
3 Contexto Tecnológico	23
1 Plataformas Móveis – <i>Smartphones</i>	23
1.1 Sistemas Operativos Móveis	24
1.2 Estatísticas e Cotas de Mercado	30

2	Principais Tecnologias Disponíveis nos <i>Smartphones</i>	34
4	Estudo do Problema	35
1	Conteúdos para o Utilizador	35
1.1	Análise dos Modelos de Negócio	36
1.2	Aplicação Quiz	38
1.3	Aplicação GeoAnúncios	41
1.4	Conclusões	44
1.5	Modelo de Negócio – GeoAnúncios	45
2	Recolha de Dados e Poupança de Energia	47
2.1	Recolha Contínua	49
2.2	Recolha Periódica	50
2.3	Controlo da Recolha através do Acelerómetro	54
2.4	Conclusões	55
5	Especificação do Sistema	57
1	Arquitetura do Sistema	57
1.1	Arquitetura – Visão Geral	58
1.2	Análise de Requisitos	58
1.3	Módulos Aplicacionais	62
1.4	Arquitetura do Sistema – Detalhe	65
2	Mensagens do Sistema de Recolha de Dados	66
6	Implementação	69
1	Definição dos Parâmetros de Desenvolvimento	69
1.1	Aplicação Móvel	69
1.2	Servidor	74
1.3	Características e Parâmetros do Sistema – Resumo	88
2	Recolha de Dados – Módulo <i>DataManager</i>	89
2.1	Recursos Utilizados e Dados Recolhidos	90
2.2	Bases de Dados	92
2.3	Recolha de Dados dos Interfaces	94
2.4	Armazenamento dos Dados	97
2.5	Envio dos Dados	97
2.6	Execução do <i>DataManager</i> como Serviço Android	101

2.7	Maximizar o Tempo de Execução do <i>DataManager</i>	102
2.8	Reconfiguração Periódica Automática	103
2.9	Controlo da Recolha de Dados	103
3	GeoAnúncios	111
3.1	Base de Dados	111
3.2	Registo, <i>Login</i> e Créditos	113
3.3	Funcionalidades	116
7	Conclusões e Trabalho Futuro	123
1	Conclusões	123
1.1	Processo de Recolha de Dados	123
1.2	Poupança de Energia	124
1.3	Segurança, Privacidade e Integridade	126
1.4	Motivação dos Utilizadores	127
2	Análise dos Resultados	129
2.1	Adesão à Aplicação	129
2.2	Dados Recolhidos	131
3	Trabalho Futuro	133
	Bibliografia	135
A	Principais Tecnologias Disponíveis nos <i>Smartphones</i> – Explicação	153
0.1	Wi-Fi – IEEE 802.11	153
0.2	Bluetooth	155
0.3	Rede Móvel Celular	157
0.4	<i>Global Positioning System</i> (GPS)	162
B	Diagramas de Classes	165
C	Protocolo de Comunicação	167
D	Aplicação GeoAnúncios	173
E	Web Site	175
F	Divulgação	177

Lista de Figuras

2.1	Modelos de negócio genéricos	21
3.1	Total de <i>smartphones</i> vendidos em 2010 e 2011 (Milhões de Unidades)	30
3.2	Vendas Mundiais de <i>smartphones</i> 1º Trimestre de 2012 por fabricante [IDC12e] .	31
3.3	Alterações na cota do mercado ano-após-ano (Percentagem) [IDC12e]	32
3.4	Vendas Mundiais de <i>smartphones</i> 1º Trimestre de 2012 por SO [IDC12a]	32
4.1	Modelo de negócio geral	36
4.2	Modelos	37
4.3	Modelo de recompensa do serviço GeoAnúncios	46
4.4	Carga elétrica utilizada por unidade de tempo num <i>smartphone</i> [Sha09]	47
4.5	Amostragem não controlada dos Interfaces	49
4.6	Estimativa da percentagem de bateria consumida por hora em recolha contínua .	49
4.7	Amostragem periódica dos Interfaces	50
4.8	Diferenças de alcances entre tecnologias	51
4.9	Estimativa da percentagem de bateria consumida por hora em recolha periódica	53
4.10	Carga elétrica utilizada pelo acelerómetro por unidade de tempo[Sha09]	54
4.11	Recolha controlada pela deteção de movimento através dos acelerómetro	55
4.12	Comparação das estimativas da percentagem de bateria consumida por hora . .	55
5.1	Servidor e dispositivo móvel	57
5.2	Arquitetura do sistema (Geral)	58
5.3	Módulos da aplicação móvel e servidor	62
5.4	Arquitetura do sistema móvel (Detalhada)	65
5.5	Arquitetura do servidor (Detalhada)	65
5.6	Mensagem de envio dos dados recolhidos	66
5.7	Configurações enviadas para a aplicação móvel	67
5.8	Informações sobre a versão atual da aplicação	67

6.1	Arquitetura da plataforma Android[ARM12]	71
6.2	<i>Web Service</i> e clientes multi-plataforma	75
6.3	Mensagens SOAP invocam processos	77
6.4	APIs REST obtêm recursos	78
6.5	Percentagem de APIs com suporte XML.[Mus11]	79
6.6	Percentagem de novas APIs com suporte JSON[Mus11]	80
6.7	Distribuição dos protocolos utilizados pelas APIs[Mus11]	81
6.8	Percentagens de websites que utilizam as diferentes linguagens de programação	85
6.9	Resultados do teste Tomcat+Axis2 Java e Apache+PHP5[STT ⁺ 08]	86
6.10	Representação das tecnologias do sistema	88
6.11	Diagrama Entidade Relação (DER) da BD dados recolhidos (Aplicação Móvel)	92
6.12	Diagrama Entidade Relação (DER) da BD dados recolhidos (Servidor)	93
6.13	Recolha de dados constante de um interface	94
6.14	Funcionamento dos <i>Timers</i> implementados	95
6.15	Tarefa de recolha e de dados executada por um <i>Timer</i>	96
6.16	Exemplo de uma tarefa de recolha e armazenamento de dados	97
6.17	Processo de envio de dados armazenados para o servidor	98
6.18	Cifragem dos dados AES-128 em modo CBC	100
6.19	Módulo GeoAnúncios e módulo <i>DataManager</i> juntos numa única aplicação	101
6.20	Módulo GeoAnúncios como aplicação e módulo <i>DataManager</i> como serviço	101
6.21	Eixos de um acelerómetro	104
6.22	Funcionamento do algoritmo de deteção de movimento	105
6.23	Valores da magnitude da aceleração linear do acelerómetro a caminhar	106
6.24	Valores da magnitude da aceleração linear com redução dos picos	107
6.25	Valores da magnitude da aceleração linear do acelerómetro de carro (telefone pousado)	108
6.26	Valores da magnitude da aceleração linear do acelerómetro de carro (telefone no bolso)	108
6.27	Mecanismo de ativação do CPU	110
6.28	DER da BD GeoAnúncios <i>Cache</i> (Aplicação Móvel)	111
6.29	Diagrama Entidade Relação (DER) da BD GeoAnúncios (Servidor)	112
6.30	Menu inicial da aplicação GeoAnúncios	113
6.31	Registo de um utilizador	114
6.32	<i>Login</i> manual de um utilizador	115

6.33	Menu principal da aplicação GeoAnúncios	116
7.1	Estatísticas de Downloads, Registos e Total de Utilizadores	129
7.2	Amostras recolhidas de cada Interface	131
A.1	Arquitetura de uma rede Wi-Fi (IEEE802.11) baseada em infraestrutura[Sch03]	154
A.2	Bluetooth <i>piconets</i> e <i>scatternets</i> [Por12]	156
A.3	Arquitetura do sistema GSM	159
A.4	Arquitetura básica de uma rede UTRAN[Sch03]	160
A.5	Posicionamento GPS	163
B.1	Módulo de Recolha de Dados	165
B.2	GeoAnúncios	166
C.1	Protocolo – Recolha de Dados (1)	167
C.2	Protocolo – Recolha de Dados (2)	168
C.3	Protocolo – GeoAnúncios (1)	169
C.4	Protocolo – GeoAnúncios (2)	170
C.5	Protocolo – GeoAnúncios (3)	171
D.1	Ecrãs da Aplicação (1)	173
D.2	Ecrãs da Aplicação (2)	174
E.1	http://geoanuncios.dsi.uminho.pt	175
F.1	Email de Divulgação – 27 de Setembro de 2012	177
F.2	Email de Divulgação – 8 de Outubro de 2012	178

Lista de Tabelas

4.1	Consumos e duração de um <i>scan</i> por interface	48
4.2	Aspetos considerados para a definição dos períodos de amostragem	52
5.1	Dados enviados (Por Interface)	66
6.1	Resumo da análise das plataformas Android e iOS	71
6.2	Configurações utilizadas para o teste[STT ⁺ 08]	86
6.3	APIs Android utilizadas para acesso às informações das Interfaces	90
7.1	Períodos utilizados na recolha de dados	131

Glossário

ACID *Atomicity, Consistency, Isolation, Durability*

ADT *Android Development Tools*

AES *Advanced Encryption Standard*

AMPS *Advanced Mobile Phone Service*

APIs *Application Programming Interface*

AP *Access Point*

BD *Base de Dados*

BSSID *Basic Service Set IDentification*

CPU *Central Processing Unit*

CRUD *Create, Read, Update and Delete*

DER *Diagrama Entidade Relação*

DSSS *Direct Sequence Spread Spectrum*

EDGE *Enhanced Data Rates for GSM Evolution*

EDI *Electronic Data Interchange*

FCC *Federal Communications Commission*

FDMA *Frequency-Division Multiple Access*

FHSS *Frequency-Hopping Spread-Spectrum*

GFSK *Gaussian Frequency Shift Keying*

GPRS *General Packet Radio Service*

GPS *Global Positioning System*

GSM *Global System for Mobile Communications*

HSPA Plus *Evolved High-Speed Packet Access*

HTML *HyperText Markup Language*

HTTP *Hypertext Transfer Protocol*

IDC *International Data Corporation*

IDE *Integrated Development Environment*

ISM *Industrial, Scientific and Medical*

JSON *JavaScript Object Notation*

LAC *Location Area Code*

LAN *Local Area Networks*

LCD *Liquid Crystal Display*

LTE *Long Term Evolution*

MAN *Metropolitan Area Networks*

MCC *Mobile Country Code*

MIDP *Mobile Information Device Profile*

MIMO *Multiple-Input and Multiple-Output*

MNC *Mobile Network Code*

OFDMA *Orthogonal Frequency-Division Multiple Access*

PHP *PHP: Hypertext Preprocessor*

QEMU *Quick EMUlator*

REST *Representational State Transfer*

RIM *Research In Motion Limited*

RSSI *Received Signal Strength Indication*

RSS *Rich Site Summary*

SC-FDMA *Single-carrier Frequency-Division Multiple Access*

SCP *Software como Produto*

SCS	<i>Software como Serviço</i>
SDK	<i>Software Development Kit</i>
SIG	<i>Special Interest Group</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOAP	<i>Simple Object Access Protocol</i>
SOP	<i>Sensors on Patrol</i>
SO	Sistema Operativo
SQL	<i>Structured Query Language</i>
SSID	<i>Service Set Identifier</i>
SSL	<i>Secure Sockets Layer</i>
SUM	<i>Sensing and Understanding human Motion dynamics</i>
SVGT	<i>Scalable Vector Graphics Tiny</i>
TCP	<i>Transmission Control Protocol</i>
TDD	<i>Time Division Duplexing</i>
TDMA	<i>Time Division Multiple Access</i>
UMTS	<i>Universal Mobile Telecommunications System</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
WLAN	<i>Wireless Local Area Network</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>eXtensible Markup Language</i>
XSD	<i>XML Schema Definition</i>

Capítulo 1

Introdução

1 Enquadramento

O Projeto SUM (Sensing and Understanding Human Motion Dynamics) no qual esta dissertação se enquadra, tem como principal objetivo o estudo do movimento das pessoas num determinado meio. O estudo do movimento humano, como a maioria das tarefas de *sensing*, está dependente do conhecimento da localização dos utilizadores ao longo do tempo.

Os *smartphones* acompanham os utilizadores na sua rotina diária. A elevada mobilidade e as tecnologias que estes dispositivos disponibilizam criam novas oportunidades de *sensing*, permitindo a utilização de tecnologias como o GPS, a Rede Móvel Celular, o Wi-Fi e o Bluetooth para a recolha de dados que permitem analisar e contextualizar o movimento da população e a sua interação com os elementos presentes no meio. Não tendo acesso às infraestruturas de rede ou aos dispositivos móveis, a recolha destes dados pode ser realizada através do uso voluntário de uma determinada aplicação por parte dos utilizadores nos seus *smartphones*. A aplicação deve ser por isso capaz de recolher dados utilizando os interfaces das várias tecnologias disponíveis, nomeadamente: a posição GPS; as redes Wi-Fi que deteta (por exemplo os dados relativos ao *Access Point* (AP), ou APs próximos); as informações sobre a rede móvel celular, através da qual é possível identificar as estações base próximas do utilizador; e os dispositivos Bluetooth nas proximidades e as suas características.

Os dados recolhidos através dos vários interfaces, devem ser enviados e armazenados num servidor para posterior processamento e análise, enriquecendo a base de dados do projeto SUM.

Um problema comum nas aplicações de *sensing* colaborativo é a participação. Por isso pretende-se motivar a utilização da aplicação através da implementação de funcionalidades úteis e atrativas, acompanhadas de um design cuidado e apelativo. Adicionalmente, o utilizador

deverá ser recompensado de alguma forma pelos dados recolhidos, abordando de forma frontal as questões que a recolha de dados levanta.

Esta abordagem de recolha de dados têm um forte impacto na autonomia energética dos dispositivos, provocando uma baixa utilização. Desta forma, o consumo de energia por parte da aplicação deverá ser minimizado, adotando algoritmos de recolha de dados que minimizem o consumo de energia.

2 Objetivos

O principal objetivo deste trabalho é conceber uma solução que transforme os dispositivos móveis pessoais (*smartphones*) em sensores móveis que permitam a observação do movimento humano no espaço físico.

Conceber esta solução corresponde a dar resposta a um conjunto de desafios, entre os quais se incluem:

- Quais as tecnologias a explorar (Wi-Fi, Bluetooth, GPS, Rede Móvel Celular GSM, Acelerómetros, Giroscópios, etc).
- Como minimizar o consumo de energia de forma a não comprometer a autonomia dos dispositivos.
- Como envolver os utilizadores, isto é, identificar funcionalidades e serviços atrativos.
- Como lidar com as questões de privacidade.
- Especificar uma arquitetura para uma aplicação de recolha de dados.
- Como desenhar e implementar uma aplicação para plataformas móveis.
- Que tipo de informação recolher de cada sensor.
- Como utilizar e recolher as informações dos sensores e interfaces do dispositivo (Wi-Fi, Acelerómetros, GPS, etc).
- Com que frequência devem ser amostrados cada um dos sensores.
- Como centralizar os dados recolhidos.
- Como divulgar a aplicação e motivar utilizadores a colaborar nos testes da aplicação.

3 Metodologia

Esta dissertação, assim como o projeto que a mesma documenta, podem ser divididos em cinco fases principais: investigação; análise; especificação; implementação; conclusão e documentação. Cada uma destas fases engloba diferentes domínios, desde da pesquisa bibliográfica e escrita, à computação móvel e engenharia de software.

Inicialmente foi feito o estudo do estado da arte, através da pesquisa e análise de projetos semelhantes, identificando os problemas, limitações e desafios, encontrados pelos diversos autores. Foi possível reunir referências bibliográficas, que permitiram firmar bases teóricas que suportam, contextualizam e reforçam a motivação desta dissertação.

Seguiu-se a identificação dos requisitos do sistema, resultando desta fase um conjunto de potenciais serviços a oferecer ao utilizador, assim como delineadas soluções para a recolha de dados de movimento.

Posteriormente, foi feita a especificação do sistema, por meio da modulação e especificação da sua arquitetura.

A fase de implementação, iniciou-se pela definição dos parâmetros técnicos e tecnológicos através do estudo das tecnologias que melhor se adequam aos requisitos do sistema. Posteriormente foi desenvolvida a aplicação móvel e o software de suporte à mesma, implementando a recolha de dados de movimento, e as funcionalidades disponível aos utilizadores. Finalmente, a aplicação foi divulgada e disponibilizada, envolvendo utilizadores reais em contextos reais.

A escrita da dissertação decorreu durante praticamente todo o período de análise e desenvolvimento. Este paralelismo simplificou e enriqueceu a documentação das várias etapas, culminando num conjunto interessante de conclusões.

4 Estrutura da Dissertação

A presente dissertação encontra-se dividida em sete capítulos. Neste capítulo introdutório é realizado o enquadramento deste projeto no domínio científico e tecnológico, sendo apresentados os objetivos e desafios inerentes.

No capítulo dois é apresentado o estudo do estado da arte, relativamente ao nível de conhecimento da comunidade de investigação na área do *sensing*. É realizada a análise da evolução dos métodos adotados, das tecnologias utilizadas e das metas alcançadas. Pretende-se no final deste capítulo, obter uma visão geral dos desafios que esta área coloca aos investigadores, de modo a que esta dissertação abranja as questões pertinentes.

No capítulo três, é feita a contextualização tecnológica, através da análise das características das principais plataformas móveis disponíveis no mercado e da comparação dos índices de utilização e de crescimento. Acresce a este capítulo, a identificação e estudo das tecnologias de ligação ao meio disponíveis nos *smartphones*, que permitem servir os objetivos de *sensing* do projeto.

O quarto capítulo, apresenta o estudo do problema, sendo delineadas as soluções para os principais desafios. Nomeadamente, em relação ao processo de recolha de dados e aos conteúdos a disponibilizar aos utilizadores através da aplicação desenvolvida.

Posteriormente, no capítulo cinco, é realizada a especificação do sistema, que serve de base à implementação. Neste capítulo são fixados os requisitos do sistema e do utilizador, sendo especificada a arquitetura do sistema.

O capítulo seis documenta o processo de implementação, definindo e apresentando os parâmetros de desenvolvimento. São discutidas as diferentes opções e analisadas as vantagens e desvantagens de cada abordagem. Este capítulo termina com a apresentação e explicação da solução implementada.

O sétimo e último capítulo expõe as conclusões obtidas através do trabalho desenvolvido. São retiradas ilações sobre os caminhos seguidos e sobre os resultados obtidos. Adicionalmente, são apresentadas algumas sugestões e orientações para um trabalho futuro.

Capítulo 2

Estado da arte

1 Objetivos do estudo do movimento humano

O movimento humano é uma área que está em constante investigação, tanto do ponto de vista científico como tecnológico. Os vários estudos focam diversos ângulos e têm objetivos distintos.

É importante começar por dividir a área de estudo do movimento humano em dois grupos: o movimento do corpo humano (biomecânica) e o movimento humano num espaço físico. A biomecânica do corpo humano é estudada pela *Kinesiology* também conhecida por *Human Kinetics*, ciência que tem por objetivo o desenvolvimento de diversas áreas, como a reabilitação física[MMRFSPJBM08], próteses mecânicas[LLC11], robôs[WMK11], assim como vídeo jogos[YCL11][PBN11] e filmes de animação[BHN04].

Por outro lado, procede-se ao estudo do movimento humano em termos de deslocação geográfica, analisando os padrões de movimento humano no espaço físico, por exemplo numa cidade, e como interagem com os objetos nesse mesmo espaço. Este projeto de dissertação insere-se neste domínio de estudo do movimento humano.

Os objetivos do estudo do movimento humano num espaço físico podem abranger diversas vertentes, como o planeamento urbano e social, publicidade e divulgação de informação, prevenção de epidemias e disseminação de vírus, aperfeiçoamento da resposta em casos de emergência e ataques terroristas[ZMvdB05], assim como o controlo do ambiente urbano, por exemplo através da medição dos níveis de ruído e poluição[AAB⁺07].

O estudo do movimento humano pode utilizar estratégias e meios distintos, sendo que os objetivos podem ser ou não similares. Desta forma o contexto do estudo é importante para definir a estratégia e metodologias adotadas.

Para estabelecer um contexto, é necessário perceber como podemos interpretar o movimento

das pessoas. Como é referido no estudo *Mobiscopes for Human Spaces*[AAB⁺07], as trajetórias humanas mostram um alto nível de regularidade espacial e temporal, podendo ser estabelecido um determinado padrão para um determinado indivíduo. O movimento geográfico, o intervalo temporal em que ocorre, e os padrões de movimento[GHB08] são aspetos importantes para analisar o movimento humano, sendo necessários por isso métodos para a obtenção de dados que permitam satisfazer estes requisitos. Para que seja possível a recolha de dados é necessário garantir:

- Participantes que constituem a amostra do estudo.
- Formas de registo do seu movimento.
- Um período de tempo mínimo para recolher dados suficientes que garantam resultados coerentes.

O primeiro grande desafio são os participantes, devido à complexidade intrínseca do Ser Humano[LH10], parte fundamental do estudo. O objetivo é maximizar tanto a amostra, ou seja o número de participantes, como o tempo do estudo, durante o qual é efetuado o registo do seu movimento. O tempo de estudo pode ser dividido no tempo total de participação de um indivíduo, por exemplo um mês, e no tempo de amostragem do seu movimento, ou seja tempo diário em que são recolhidos dados, sendo que durante o tempo de amostragem a recolha de dados é influenciada também pela frequência de amostragem, que representa a frequência com que é registada a posição da pessoa.

Maximizar o tempo de estudo e a amostra, aumenta a coerência dos resultados obtidos. No entanto, também o tempo e a frequência de amostragem têm influência nos resultados. Se o tempo de amostragem for reduzido ou a frequência de amostragem baixa, a quantidade de dados para processar é menor, não sendo possível por isso obter uma resolução pormenorizada e correta do movimento da população. Neste contexto, os equipamentos utilizados, e a forma como é registado o movimento das pessoas, desempenham um papel crucial nos resultados.

Existem diversas limitações em relação ao equipamentos, nomeadamente os custos que limitam o número de dispositivos disponíveis, e consequentemente o número de participantes, o que resulta na redução dos dados recolhidos.

A utilização de redes de sensores para a recolha de dados do meio urbano e da população tem vindo a crescer, sendo objeto de investigação em vários projetos. Este método de recolha de informação é denominado de *Sensing*, quando aplicado a um meio urbano é geralmente denominado de *Urban Sensing*.

2 *Sensing* Colaborativo

A primeira geração de redes de sensores é composta por sensores fixos, que permitem a recolha de informações do meio que os rodeia. Com o progresso tecnológico, as redes de sensores têm vindo a ser utilizadas em múltiplas áreas, desde monitorização do meio ambiente (por exemplo da qualidade do ar[erg12]), aplicações para a saúde pública e caracterização do tráfego[AAB⁺07][HBZ⁺06], assim como em aplicações militares para deteção de ataques biológicos, químicos e nucleares[ZMvdB05].

Contudo, através de sensores fixos é difícil a cobertura de grandes áreas urbanas. Os custos de aquisição, instalação e manutenção de dispositivos suficientes é um enorme entrave à escalabilidade do sistema. Não é possível, por isso, a colocação de dispositivos sensoriais com densidade espacial suficiente para obtenção de amostras quer permitam uma visão geral do meio. A amostragem fica restrita a pontos fixos, acrescentando ainda a necessidade de uma infraestrutura de rede complexa para reunir os dados recolhidos para análise.

Devido a estas limitações, o foco das pesquisas de redes de sensores evoluiu das redes estáticas para redes que se adaptam às alterações dos meios urbanos, numa abordagem centrada nas pessoas e na sua mobilidade[LEM⁺08], dando origem a arquiteturas baseadas em redes de sensores móveis.

Tendo esta abordagem por base a população humana e a sua mobilidade no meio urbano, está dependent e da colaboração dos participantes, sendo este método de *sensing* geralmente referido como colaborativo. As vantagens são numerosas, com menos sensores é possível cobrir maiores áreas dinamicamente[ZMvdB05], podendo ainda ser utilizados sensores móveis e sensores fixos de forma complementar.

Abdelzaher et al.[AAB⁺07], apresentam um exemplo comparativo entre uma arquitetura baseada em sensores fixos e uma baseada em sensores móveis. Considerando a análise da qualidade do ar de uma cidade, onde são necessários espectrómetros de massa de custo elevado para a medição de poluentes, torna-se extremamente dispendiosa uma abordagem baseada em infraestruturas de sensores fixos. No entanto, se os sensores forem aplicados em nós móveis, como por exemplo carros, é possível cobrir uma área muito superior tornando a análise mais abrangente e minimizando os custos.

Vários desafios surgem com a utilização de sensores móveis, devido à sua elevada dependência das pessoas e da sua movimentação.

Podemos classificar os sensores móveis em dois grupos[MTZ05]: controlados e não controlados. Sensores controlados são dispositivos em que a sua movimentação pode ser controlada ou facilmente prevista, de modo a cobrir uma determinada área que permita concluir uma tarefa de *sensing*. Um exemplo são os sensores aplicados em comboios e autocarros, que mantêm sempre os mesmos trajetos. Sensores transportados por pessoas, aplicados em automóveis, ou em outros nós móveis onde não é possível controlar ou prever a sua movimentação, são denominados de sensores móveis não controlados. Apesar dos desafios, a viabilidade da utilização de sensores móveis não controlados têm vindo a ser comprovada[MTZ05].

Têm surgido projetos interessantes que utilizam sensores móveis para obter informações sobre o meio urbano. Em 2005, foi apresentado na conferência *Military Communications* o projeto *Sensors on Patrol* (SOP)[ZMvdB05] que permite a deteção de ameaças químicas e biológicas, e em 2006 o *CarTel Project*[HBZ⁺06] apresentou uma solução que utiliza nós móveis para aplicações como monitorização do tráfego e condições das vias.

Existem assim muitos sensores com enorme potencial espalhados pelos meios urbanos, no entanto os dispositivos móveis pessoais (telemóveis, *smartphones* e *tablets*) são os que melhores perspetivas oferecem, e por isso o foco de investigação tem vindo a direccionar-se neste sentido.

A rápida evolução tecnológica destes dispositivos, tem vindo a criar novas oportunidades para o *sensing* colaborativo. Estes dispositivos estão equipados com inúmeras tecnologias, sensores e interfaces que podem, sem dúvida, ser explorados e graças às suas múltiplas tecnologias de conectividade são capazes de enviar esses dados de forma mais simples, rápida e com menores custos que os outros equipamentos. O número massivo deste tipo de dispositivos num meio urbano, e a sua elevada mobilidade, são mais valias adicionais e muito promissoras.

Porém surgem desafios com a utilização dos dispositivos móveis pessoais, desde a segurança e privacidade dos utilizadores, à angariação de pessoas dispostas a colaborar. Cativar utilizadores para participarem neste tipo de sistemas não é uma tarefa simples, sendo um dos desafios mais complexos e difíceis de ultrapassar pela comunidade de investigação, sendo um ponto que tem atraído a sua atenção. Em vários estudos publicados[LEM⁺08][LH10][KKT09][par10], os autores dividem o *sensing* colaborativo em dois grupos dependendo da metodologia utilizada: participativo e oportunista; e discutem os prós e contras das abordagens participativa e oportunista.

2.1 *Sensing* Oportunista e Participativo

O papel assumido pelas pessoas nos sistemas de *sensing* colaborativo, tem grande impacto neste tipo de sistemas, tanto em termos de escalabilidade, como na diversidade das aplicações que é possível suportar[LEM⁺08].

Utilizar os dispositivos móveis da população como bloco fundamental do sistema de *sensing* implica que os proprietários influenciam a arquitetura do sistema resultante. Lane et al.[LEM⁺08] aborda esta questão, referindo que é necessário perceber em que medida os utilizadores estão dispostos a participar em sistemas de larga escala, e de que forma devem ser participantes ativos e conscientes da importância do cumprimento dos requisitos da aplicação. Dependendo do nível de envolvimento do participante no sistema, este pode ser baseado em *sensing* participativo ou oportunista.

No *sensing* participativo, o utilizador é parte ativa na recolha de informação, existindo uma grande interação. Neste caso, classificamos o participante de "consciente ativo". Se um participante que utiliza uma aplicação e manualmente inicia a recolha de informações sobre o meio que o rodeia, ou sobre si próprio, submetendo depois essas informações, estamos perante *sensing* participativo.

No *sensing* oportunista, o participante desempenha um papel passivo na recolha da informação, sendo que a interação necessária para a recolha de informação é mínima ou inexistente. Pode no entanto existir outro tipo de interatividade, mas da qual a recolha de dados não está dependente. Neste caso o participante é "consciente passivo". Neste modelo, a recolha e envio da informação é feita quando existem condições propícias. A recolha de dados da utilização de uma rede Wi-Fi, ou a recolha de dados automática e periódica através de uma aplicação instalada no *smartphone* de um utilizador, são exemplos de *sensing* oportunista.

É importante deixar claro que, em ambos os casos, o participante está consciente e tem conhecimento dos objetivos de *sensing*, devendo os responsáveis pela recolha dos dados consciencializar os participantes sobre todo o processo.

Em [LEM⁺08], é estabelecido um modelo que permite avaliar os dois métodos, tendo por base um cenário em que são utilizados dispositivos móveis pessoais. Este modelo avalia a probabilidade de sucesso de uma estratégia participativa em que é feito um pedido de recolha de dados de um determinado sensor do dispositivo ao utilizador. Os autores constatarem que à partida uma abordagem participativa, onde existe grande envolvimento das pessoas, consegue superar a oportunista. No entanto, se é exigida elevada interação com o utilizador para a recolha dos dados, o seu interesse vai reduzindo ao longo do tempo, e desta forma a abordagem

oportunista supera a participativa em estudos de longa duração.

Os autores consideram que as duas abordagens podem ser complementares em alguns cenários, referindo contudo que utilizando uma abordagem oportunista é possível mais facilmente suportar sistemas de maior escala e diversidade.

3 Dispositivos Móveis Pessoais como Sensores

A utilização dos dispositivos móveis pessoais, como nós de uma rede de sensores móvel colaborativa, permite o aparecimento de novas arquiteturas para outros tipos de aplicações.

Este tipo de dispositivos permitem a criação de uma rede de sensores poderosa, sem necessidade de grandes investimentos em *hardware* e *software* uma vez que as pessoas transportam os seus próprios sensores pelo meio urbano contribuindo com dados para o sistema[SPG]. Os dispositivos móveis pessoais partilham algumas das vantagens dos outros sensores móveis, nomeadamente:

- Área de cobertura superior, com menos dispositivos em relação às redes de sensores fixos.
- É possível dispensar a criação de infraestruturas de rede dedicadas de suporte para a recolha da informação.
- Arquiteturas mais económicas e dinâmicas.
- Permitem aplicações centradas nas pessoas e na sua movimentação, recolhendo informação não só do meio urbano mas também aspetos da vida das pessoas.

A abordagem baseada em dispositivos móveis pessoais permite aumentar de forma razoável a quantidade de dados recolhidos sem custos adicionais. Além disso não é necessária a manutenção dos dispositivos, e se alguns dispositivos deixarem o sistema, porque não são controlados, surgirão novos. A limitação do tempo de duração da amostragem é também eliminada, deixando de estar restrita a um tempo pré-definido ou ao tempo de vida dos sensores. Torna-se ainda mais simples participar, não é necessário transportar um equipamento específico, o participante já utiliza e transporta o seu dispositivo diariamente.

Adicionalmente, os dispositivos móveis pessoais facilmente superam os outros sensores móveis no que diz respeito à conectividade. Projetos desenvolvidos anteriormente que utilizam sensores móveis aplicados em carros[HBZ⁺06], mostram que existe um elevado número de pontos de acesso Wi-Fi sem segurança que permitem o envio dos dados. Os pontos de acesso Wi-Fi

sem segurança são sem dúvida uma boa solução, no entanto os dispositivos móveis pessoais não ficam dependentes deste método de comunicação. A maioria dos *smartphones* no mercado estão equipados com várias tecnologias de comunicação, como o Wi-Fi, serviço de dados da rede móvel celular (*General Packet Radio Service* (GPRS), HSPA+, *Long Term Evolution* (LTE)), e Bluetooth, que podem funcionar tanto para efeitos sensoriais, como de interfaces para o envio dos dados recolhidos.

A maioria dos serviços disponíveis nos *smartphones* requerem ligação à Internet, e a maioria são massivamente utilizados (por exemplo as redes sociais e o email), criando oportunidades de comunicação para envio dos dados recolhidos, sendo este método de comunicação denominado de comunicação oportunista.

Uma grande desafio que surge nos sensores móveis em geral é a capacidade da bateria, que limita o seu tempo de funcionamento. Os dispositivos móveis pessoais são carregados sempre que necessário pelos seus utilizadores, garantindo o seu constante funcionamento. No entanto, o elevado consumo de energia por parte de uma aplicação de *sensing*, num *smartphone*, sentença quase de imediato a mesma ao insucesso. Os utilizadores apreciam a sua autonomia e por isso a do seu dispositivo. Se um utilizador necessitar de recarregar constantemente o seu *smartphone* devido à utilização de uma aplicação, poderá deixar de a utilizar.

Este desafio, aliado aos desafios de segurança e privacidade, leva-nos mais uma vez à questão da motivação da população para participar utilizando o seu dispositivo móvel pessoal. Resolver estas questões é importante para que as pessoas não percam o interesse pela aplicação.

3.1 Segurança, Privacidade e Integridade

Na geração anterior de redes sem fios, nomeadamente nas de sensores fixos as questões de segurança e integridade, por exemplo a segurança da infraestrutura de rede contra ataques maliciosos e a garantia da integridade dos dados, são desafios mais críticos do que a privacidade. Estes e outros problemas são descritos em diversas publicações que apresentam algumas soluções para redes de sensores fixos [PSW⁺01][Zhu03] [YHSG03][TKFH06].

No entanto, a utilização dos dispositivos móveis pessoais como sensores para modelos de *sensing* colaborativo introduz novos desafios de segurança, privacidade e integridade. Apesar de alguns estudos serem direcionados para estes problemas[CEL⁺06][PBB09], na maioria dos projetos da área, que utilizam dispositivos móveis pessoais ou dispositivos móveis específicos para a recolha de informação, estes problemas são apenas referenciados ou abordados teóricamente. A maioria dos sistemas de *sensing* não integram mecanismos eficientes que permitam ultrapassar estas questões.

Os *smartphones* estão diretamente ligados com aspetos da vida pessoal dos seus utilizadores. Contêm um grande volume de informação pessoal e podem ser um apontador constante da sua localização, por isso é necessário garantir a segurança e privacidade. Kapadia et al.[KKT09] aborda estes problemas do ponto de vista do *sensing* oportunista centrado nas pessoas e nos seus dispositivos, descrevendo desafios e abordando soluções possíveis. Shin et al. apresenta o sistema AnonySense[SCP⁺11], que os autores descrevem como sendo um sistema *privacy-aware*, para a criação de aplicações baseadas em *sensing* colaborativo oportunista, em dispositivos móveis pessoais.

Outras publicações abordam também os problemas de privacidade das pessoas externas aos sistemas de *sensing* (*Bystanders*)[SPG], que apesar de não serem participantes ativos, a sua privacidade pode ser comprometida, consequência por exemplo de uma tarefa de *sensing* em que um participante utiliza a câmara do seu *smartphone* para recolher uma imagem de um determinado local a uma determinada hora.

Em relação à segurança é necessário garantir que os dados recolhidos são protegidos tanto no processo de envio, como no seu armazenamento. Se os dados não forem protegidos podem ser interceptados por terceiros, colocando em risco a privacidade e segurança do seu fornecedor, revelando informações ou a localização do utilizador. Para garantir a segurança, podem ser adotados mecanismos criptográficos e ligações seguras.

A privacidade dos participantes é difícil de proteger. Por mais restrita que seja a informação recolhida, é sempre necessário recolher informação suficiente para cumprir os requisitos e contextualizar os dados recolhidos. Por exemplo a recolha da localização do participante deve ser acompanhada da data e hora para que seja contextualizada numa linha temporal.

O anonimato pode ser utilizado como mecanismo de privacidade. Como é referido em [LEM⁺08], em muitos casos os dados recolhidos são muito mais importantes do que a identificação da sua fonte. Por exemplo o sistema AnonySense[SCP⁺11] baseia-se neste conceito. No entanto não é fácil garantir o total anonimato dos participantes, muitas aplicações requerem certas informações como por exemplo o número de identificação do dispositivo ou o seu endereço MAC para correlacionar os dados e unificar a sua fonte.

A integridade dos dados pode ser facilmente comprometida, porque os dados são recolhidos de dispositivos controlados pelos seus utilizadores. Além disso, ao serem utilizadas formas de comunicação oportunista para o envio dos dados, nomeadamente através de pontos de acesso Wi-Fi desprotegidos, pode existir adulteração dos dados por agentes maliciosos. No caso de existir uma forma de recompensa para os participantes que submetem dados para o sistema, poderá existir ainda a tentativa de fabricação de dados falsos para obtenção da recompensa,

colocando em risco a correção dos resultados obtidos através dos dados.

4 Projetos de *Sensing*

Existem projetos dedicados ao desenvolvimento do *sensing* urbano, criando sistemas que têm por base as pessoas e a sua mobilidade para o estudo de vários aspetos relacionados com a comunidade e com o meio urbano. Alguns destes projetos investigam como podem ser os dispositivos móveis pessoais utilizados para a recolha e partilha de informações, tendo já sido desenvolvidas algumas soluções. Vejamos alguns exemplos de projetos já desenvolvidos nesta área.

■ Mobiscopes

Mobiscopes[AAB⁺07] estende o modelo tradicional de redes de sensores, introduzindo desafios na gestão, integridade e privacidade dos dados, e no design do sistema. Os autores descrevem *Mobiscope* como um conjunto de sensores móveis distribuídos em sistemas capazes de efetuar tarefas de *sensing* e que permite atingir uma cobertura sobre uma amostragem de alta densidade cobrindo uma ampla área através da mobilidade. São descritas várias classes de *Mobiscopes*, como por exemplo os veiculares, sendo apresentados os requisitos comuns e os desafios inerentes aos sistemas. Neste estudo são abordados vários projetos como o CarTel Project[HBZ⁺06] e o Nokia Sensor Planet[Res10].

■ Cartel Project

O projeto CarTel[HBZ⁺06] consiste num sistema de sensores móveis computadorizado para recolher, processar, enviar e visualizar dados de sensores localizados em unidades móveis, como por exemplo automóveis. Os sensores instalados nos veículos permitiram recolher padrões de tráfego em Boston e Seattle, assim como dados de localização, direção, velocidade e aceleração, entre outras informações do veículo e meio urbano.

Os nós utilizam comunicação oportunista, utilizando as ligações sem fios desprotegidas encontradas aleatoriamente para se conectarem à Internet, ou enviando os dados para nós próximos através de Bluetooth. Existe uma interface *Web* que permite o acesso à base de dados e permite efetuar configurações da recolha de dados, como por exemplo o período de amostragem. Este projeto permite aplicações como monitorização de tráfego, diagnóstico do

veículo e das condições da via. É um projeto que se baseia na utilização que as pessoas fazem dos seus automóveis.

O sistema foi implementado em seis automóveis durante o período de um ano. Durante este estudo não foram consideradas formas de preservar a privacidade dos dados recolhidos. No entanto, mais tarde foram efetuados estudos por elementos do MIT Cartel com propósitos de proteger a privacidade dos condutores em sistemas de localização veicular [PBB09], sendo utilizados protocolos criptográficos para proteger os dados.

■ **Nokia *Sensor Planet***

O Nokia *Sensor Planet*[Res10] é uma iniciativa da Nokia para a criação de uma plataforma de testes e de pesquisa de redes sem fios globais, centradas em dispositivos móveis. Alguns dos objetivos desta iniciativa são tornar a investigação deste tipo de redes de sensores mais simples, tornando possível a grupos individuais de investigação desenharem e implementarem experiências de larga escala. Esta iniciativa pretende ainda estabelecer uma comunidade (*Mobile Wireless Sensor Network Open Source Community*) que permita a partilha de código, acelerando a inovação de redes de sensores sem fios orientadas ao consumidor, e a criação de um ecossistema para a colaboração e desenvolvimento industrial e académico.

Durante 2006, vários projetos foram desenvolvidos, como por exemplo o Nokia *Remote Sensing platform* (NORS), e o *Virtual Graffiti* desenvolvido pelo Nokia *Research Center* (NRC) Toijala.

O Nokia *Remote Sensing platform* (NORS) é baseado na Nokia *Remote Sensing Architecture* (N-RSA), desenvolvida internamente e especificada para soluções de *sensing* remoto baseadas em telefones móveis. O NORS fornece uma plataforma para *mobile-centric wireless sensing*, implementada através de um servidor central e dispositivos móveis atuando como *gateways* para *Wireless Sensor Networks* locais. A solução para dispositivos móveis é implementada em Java MIDP2.0.

O NORS permite a fácil extensão da plataforma atual, estimulando a inovação a nível da aplicação, sendo utilizado num projeto da Universidade de Cambridge para a monitorização da poluição. Neste projeto, ciclistas e pedestres recolhem dados ambientais a partir de sensores móveis e estacionários, transferindo os dados em tempo real para um servidor, que por sua vez fornece alertas para a comunidade móvel. Também o MIT MediaLab e outras universidades norte-americanas manifestaram interesse em utilizar o projeto.

O Virtual Graffiti é uma aplicação atualmente em desenvolvimento no NRC Toijala. O movimento dos utilizadores é registado e eles podem deixar *tags* virtuais no meio urbano.

Estas *tags* podem incluir fotos, música, vídeo ou texto. Outros utilizadores podem visualizar e fazer comentários sobre os *tags*.

Outro dos projetos desenvolvidos foi o Manhattan *Story Mashup*, que consistiu num jogo colaborativo que envolveu jogadores de Manhattan e da Web. O MSM fez parte do *Urban Games Festival* em 2006. O jogo permitia aos utilizadores Web colaborarem na criação de histórias e os jogadores em Manhattan utilizavam os seus telefones para tirar fotografias que ilustravam as histórias, sendo estas apresentadas nas telas de Times Square. Esta iniciativa teve elevado sucesso, permitindo a recolha de vários tipos de dados.

Existem ainda diversos desafios nos quais o Nokia *Sensor Planet* reúne esforços: como tornar o grande volume de dados dos sensores útil, como gerir os dados, como utilizar os dados para auto-configuração e adaptação dos dispositivos, como integrar (em ambos os sentidos) as redes de sensores sem fios para serviços baseados na Internet, e como gerir o consumo de energia dos nós.

■ Urban Atmospheres

O Urban Atmospheres[urb12] é um projeto conjunto da UC Berkeley e da Intel *Research* para o desenvolvimento de aplicações de avaliação de várias áreas, desde qualidade ambiental à comunidade e urbanismo. Os focos de pesquisa deste projeto abordam vários sub-temas, incluindo:

- Lugar: Qual é o significado dos vários locais públicos? Que exemplos usamos para interpretar os locais e como a computação urbana altera a nossa perceção dos vários locais?
- Comunidade: Quem são as pessoas com quem partilhamos a cidade? Como é que elas influenciam a paisagem urbana?
- Como podem ser os edifícios, metros, calçadas, parquímetros e outros componentes físicos convencionais da paisagem urbana utilizados por ferramentas e tecnologias emergentes?
- Qual é o caminho a seguir por uma cidade utilizando as novas ferramentas urbanas? Como irá a navegação e o movimento, numa cidade inteira, ou num pequeno espaço urbano, ser influenciados pela introdução da tecnologia de computação urbana?

O Urban Atmospheres foca esforços na investigação do urbanismo participativo e explora uma visão centrada nas pessoas, partilhando, e discutindo o meio envolvente e a qualidade do ar utilizando dispositivos móveis pessoais, através de por exemplo diversos sensores anexados para a medição da qualidade do ar. O Ergo[erg12] é um sistema simples que permite a qualquer

pessoa com um telefone móvel rapidamente explorar, consultar, e aprender sobre a qualidade do ar num determinado local. Os utilizadores podem receber em tempo real as medições da qualidade do ar de uma determinada cidade dos Estados Unidos. Para isto enviam uma SMS para um determinado número e recebem também por SMS o resultado. Os utilizadores podem consultar as medições para os diversos locais, podendo observar as mudanças ao longo do dia e as previsões para os próximos dias.

Outro projeto no âmbito do Urban Atmospheres, é o Citizen Science[Sci12] que pretende estudar e construir uma série de dispositivos móveis equipados com novos sensores, e também uma infraestrutura que permite a partilha e centralização das medições dos sensores pessoais. Este projeto tem como meta global de longo prazo desenvolver novos paradigmas de comunicação que permitam às comunidades produzirem informações credíveis, que possam ser compreendidas pela população em geral. Recentemente, através da colaboração com a cidade de San Francisco, este projeto colocou sistemas de deteção da qualidade do ar na frota municipal de veículos de limpeza das ruas, permitindo aos dispositivos recolher rua-a-rua leituras da qualidade do ar, fornecendo tanto cobertura extensiva como sistemática da cidade.

Outro projeto no espírito da computação e *sensing* urbana, é o *Participatory Urbanism*[par10] que permite a partilha de novas ou já existentes tecnologias urbanas, envolvendo a participação do cidadão individual. O *Participatory Urbanism* promove novos estilos e métodos para o cidadão se tornar pró-ativo no seu envolvimento com a cidade. Exemplos de *Participatory Urbanism* incluem fornecimento de kits e ferramentas de *hardware* móveis para pessoas comuns para que desta forma se tornem autores de novos objetos urbanos, gerando necessidades individuais e coletivas baseadas em ferramentas de diálogo em torno da utilização desejada dos espaços verdes urbanos.

■ MetroSense

O projeto MetroSense[Met12] trabalha para desenvolver novas aplicações, técnicas de classificação e abordagens de privacidade, assim como paradigmas de *sensing*, permitindo uma rede de sensores móvel global na qual os sensores são transportados pelas pessoas durante o seu dia. No âmbito deste projeto já foram desenvolvidas algumas soluções, como o CenceMe[MLEC07] que permite a recolha e a partilha nas redes sociais de informações da atividade pessoal. Essas informações incluem o estado do utilizador do ponto de vista de atividade física, o seu estado de humor, os seus hábitos, e o que o rodeia, como por exemplo o nível de ruído.

Outro projeto é o AnonySense que se apresenta como uma arquitetura ciente das questões da privacidade para a criação de aplicações baseadas em *sensing* colaborativo e oportunista,

utilizando dispositivos móveis pessoais. Permite o envio de tarefas de *sensing* que são distribuídas pelos dispositivos de participantes anônimos, recebendo os dados dos dispositivos. É um dos poucos projetos que tenta lidar com os desafios da privacidade introduzidos pelos modelos oportunistas, utilizando o anonimato como forma para proteger a privacidade dos participantes.

O SoundSense[LPL⁺09] tenta explorar as capacidades do microfone dos dispositivos móveis. Sendo possível fazer inferências sobre a atividade humana, localização e eventos sociais através do som. É um sistema para *sensing* através de som sendo desenhado para funcionar em telefones com recursos limitados.

O BikeNet[EML⁺07] é um sistema de *sensing* móvel para ciclistas. Permite aos ciclistas compartilharem informações sobre eles próprios e sobre os caminhos que percorrem, para visualização em tempo real. As bicicletas estão equipadas com sensores que comunicam com nós próximos e com a infraestrutura de rede.

■ LifeMap

LifeMap[CC11] é uma aplicação implementada para dispositivos Android, que funciona como gerador de contexto através dos sensores dos *smartphones*.

Esta aplicação utiliza os sensores inerciais do *smartphone* para, em ambiente fechados, fornecer informações de localização. Essa informação é combinada com a de posicionamento GPS e Wi-Fi de forma a gerar contexto da vida diária do utilizador. O sistema não necessita de uma infraestrutura, ou da utilização de *hardware* dispendioso, e utiliza uma técnica baseada em eventos que reduz o consumo de energia, utilizando um conjunto mínimo de sensores para definir o contexto de uma dada situação.

O objetivo principal do projeto LifeMap é ser capaz de determinar a localização em ambientes fechados com precisão a nível de compartimentos de um habitação ou edifício, construindo um mapa de contexto sobre o mapa geográfico. A aplicação LifeMap gera contextos da vida do utilizador categorizados em quatro partes:

- Localização: A posição geográfica do utilizador é especificada através das propriedades dos pontos de acesso Wi-Fi detetados, de modo a identificar a posição com precisão a nível de compartimentos.
- Atividade: Definida pelo movimento do utilizador e pela utilização do *smartphone*. O movimento do utilizador inclui andar de carro, a pé, a correr, e parado. A utilização do *smartphone* inclui envio de mensagens, chamadas, tirar fotografias, e navegar na Web.
- Conetividade: Mostra o estado atual da ligação de rede, do GSM ou do Wi-Fi.

- Meio envolvente: É um conjunto de circunstâncias em volta do utilizador.

O conceito chave deste projeto é a sua capacidade de registar a localização em meio fechados utilizando os sensores inerciais, e a agregação de localizações idênticas (por exemplo dentro do mesmo compartimento) utilizando pontos de acesso Wi-Fi. Na abordagem adotada, o processo de agregação refina a informação de localização baseando-se em dados históricos, o sistema identifica um local idêntico através da identificação do compartimento.

Para o registo de movimento em ambientes fechados, são utilizados o acelerómetro e a bússola digital do *smartphone*. Quando um utilizador está em movimento, é recolhida informação de ambos que é utilizada para aproximar a posição atual do utilizador.

Os utilizadores desta aplicação podem visualizar a informação detalhada sobre os seus pontos de interesse no Google Maps e efetuar o *upload* dos seus contextos, ou o *download* de contextos de outros utilizadores.

5 Formas de Recompensa

Consolvo et al.[CSM⁺05] verificou que quando é solicitada a partilha da sua localização, os utilizadores querem saber porque é que essa informação é necessária.

Na maioria dos casos, os utilizadores de um sistema de *sensing* colaborativo permitem que o seu dispositivo sirva os propósitos do sistema sem beneficiarem de qualquer tipo de recompensa. Os utilizadores consomem os recursos dos seus dispositivos, como por exemplo a bateria, podendo ainda comprometer a sua privacidade expondo a sua localização ou informações pessoais. Deste modo é difícil motivar as pessoas a participarem, além de que a probabilidade dos participantes deixarem o sistema porque não existe algo que recompense o seu esforço é elevada. Dependendo dos investimentos que o utilizador faz, seja em termos de consumos de recursos do seu dispositivo móvel ou a custo da sua privacidade, a recompensa deverá ser suficientemente satisfatória. O propósito dos mecanismos de recompensa é atingir e fidelizar um adequado número de participantes, prevenindo abandonos. O incentivo necessário para manter os participantes não é linear, pode variar de participante para participante, e depende também do tipo de dados recolhidos, sendo por isso complicado inferir um incentivo ótimo comum para este tipo de aplicações.

Em[LEM⁺08] é feita a exposição de alguns problemas do *sensing* participativo, concretamente quando é necessária a interação intensiva com o utilizador, no entanto não é abordada a possibilidade da criação de um modelo de negócio que minimize esses problemas.

Aplicações de *sensing* participativo têm diferentes abrangências segundo o tipo de monitorização que realizam, podendo efetuar a monitorização do meio envolvente e/ou monitorização pessoal, por exemplo da atividade do utilizador. Quando existe monitorização da atividade pessoal é mais difícil motivar a população a participar devido ao receio do utilizador em relação à sua privacidade, principalmente quando o participante não têm nada a ganhar com isso. Em [LH10] é abordada esta questão, são apresentados os pontos críticos de uma abordagem participativa e como podem ser minimizados através de incentivos financeiros, por exemplo através da venda dos dados por parte dos participantes. Este tipo de abordagem permite reduzir a desistência e salvaguardar o sistema das questões de privacidade dos participantes. Neste caso, os utilizadores percebem que apesar da sua privacidade estar em risco, estão a ser recompensados financeiramente por isso, no entanto se o valor da recompensa for demasiado baixo pode não despertar interesse nos utilizadores. Além disso, saberem que estão a ser recompensados financeiramente pode não ser suficiente, porque se não existirem garantias de privacidade, o receio dos utilizadores normalmente é mais forte. Um dos problemas em muitos sistemas é que o seu funcionamento não é claro para os utilizadores, não sendo explicados para que fim são utilizados os dados e por quem.

É necessário por isso que esteja especificado um modelo de negócio claro que motive os participantes, evidenciando as vantagens da participação e a necessidade da recolha dos dados. O modelo deverá permitir aos participantes visualizarem de forma conceptual todo o sistema de *sensing* no qual participam e desta forma criar um nível de confiança. Ganhar a confiança dos participantes é um dos desafios mais complexos, mas essencial. Um modelo de negócio coeso, simples e claro, que seja recompensador para os participantes é prioritário para o sucesso da aplicação à qual o mesmo servirá de base.

5.1 Modelos de Negócio

Um modelo de negócio, em termos gerais, descreve como duas entidades relacionam os seus interesses económicos, sociais ou outro tipo de valores. A criação de um modelo de negócio faz parte da definição da estratégia de negócio. Em [Pop11] o autor refere que formalmente o modelo de negócio possui três características: o tipo de recompensas ou serviços prestados, o tipo de modelo e o modelo de receita. Descreve que o tipo básico de recompensas são: recompensas financeiras ou físicas (produtos), recompensas intangíveis como é o caso do software e da propriedade intelectual, assim como serviços humanos, neste caso tempo e esforço humano.

O modelo de receita é descrito em [Pop11] como sendo a forma de definir como a entidade é compensada pelos serviços e produtos que fornece. O autor aborda ainda a abrangência do

modelo de negócio de software, que dependente dos serviços e funcionalidades oferecidas.

Em [Kak03] é referido que a importância da indústria de software tem aumentado por diversos fatores, entre eles o crescimento de softwares embebidos que incluem serviços. Os modelos de negócio de software têm como base a comercialização de um software ou aplicação, ou de serviços prestados através dos mesmos. Em sectores de evolução rápida, como é o caso da indústria de software, os modelos de negócio são um elemento chave para o acelerar da difusão da inovação. Em [SB12] é apresentado uma *framework* do modelo de negócio para a indústria de software, que é composto por 20 elementos que estão agrupados em 5 grupos. Estes elementos estão relacionados com a estratégia adotada, com a receita esperada, com os princípios técnicos, com a indústria e consumidores alvo, e com a sua utilização.

Em [Kak03] são abordados os serviços baseados em localização, sendo dado o exemplo de um serviço que permite encontrar um cinema que esteja a exibir um determinado filme. Este serviço é prestado ao cliente através de dispositivos móveis como os *smartphones*, que acedem a sistemas com hardware e software complexos (servidores), no entanto estes sistemas são transparentes para o utilizador.

Os modelos de negócio de software podem ser diferenciados dependendo se o *software* é comercializado como produto, como serviço, ou uma combinação dos dois. No *Software* como Produto (SCP), o *software* em si é o elemento chave do modelo de negócio, o consumidor paga pelo direito a uma cópia e muitas vezes pela manutenção e suporte. No caso *Software* como Serviço (SCS) o núcleo do software não é entregue ao cliente, no entanto o cliente tem acesso às suas funcionalidades, por exemplo através da subscrição do serviço [Pop11].

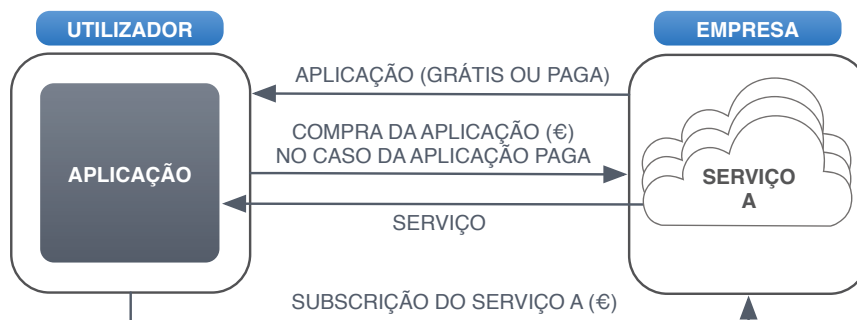
Como referido anteriormente, o modelo de receita define como a entidade que oferece um produto é compensada, existindo normalmente retorno individual por cada serviço ou cópia de um software. No entanto, no caso do SCS isso pode não se verificar. O utilizador poderá subscrever uma combinação de serviços e em troca existe um retorno para a entidade prestadora dos mesmos.

No caso das aplicações para dispositivos móveis o princípio é semelhante. O modelo de negócio, mais especificamente o modelo de receita, pode ser baseado na venda da aplicação, como por exemplo um jogo que é comprado e descarregado para o *smartphone* (Fig.2.1(b)), ou pode ser baseado num serviço ou conjunto de serviços fornecidos através da aplicação. Algumas aplicações são grátis e permitem aceder a um serviço grátis, existindo outros serviços extra pagos que o utilizador pode subscrever (Fig.2.1(a)). Em alguns casos existe a junção dos dois modelos, como é o caso de algumas aplicações de navegação GPS, a aplicação é paga e normalmente inclui alguns mapas, a empresa fornece o serviço de navegação e podem ser

subscritos serviços extra, como por exemplo os alertas do estado do tráfego.



(a) Modelo de Negócio Aplicação como Produto



(b) Modelo de Negócio Aplicação como Serviço

Figura 2.1: Modelos de negócio genéricos (a),(b)

Em [Pop11] é dado o exemplo da Google que foca os seus esforços na criação de produtos SCS para oferecer aos seus utilizadores, como por exemplo o Gmail. A receita da Google vem da publicidade que é conseguida através do seu ecossistema de SCSs, mas também dos dados e estatísticas obtidas na utilização do seu ecossistema de serviços e graças a isto a Google facilita a relação oferta-procura. Este modelo de negócio tem duas vantagens, primeiro consegue informação sobre as pesquisas dos clientes sem custos, segundo vende o espaço de publicidade enquadrado nas necessidades do utilizador com recurso às informações que possui.

Se o modelo for baseado num serviço ou aplicação útil, interessante e grátis, a utilização do serviço irá massificar-se progressivamente, sendo que essa massificação irá dissipar as preocupações dos utilizadores com a sua privacidade. Se observarmos o exemplo do Facebook e do Google, percebemos facilmente que a maioria das pessoas utilizam estes serviços sem se preocuparem se a sua privacidade é ou não salvaguardada porque, além destes serviços serem úteis e grátis, a elevada utilização tem um efeito psicológico muito comum – se todos utilizam eu não tenho motivos para me preocupar – sendo assim criado um nível de confiança.

Capítulo 3

Contexto Tecnológico

1 Plataformas Móveis – *Smartphones*

Até há poucos anos atrás, um telemóvel tinha como função a realização de chamadas de voz e envio de mensagens, permitindo mais tarde o acesso através de WAP (*Wireless Application Protocol*) a conteúdos disponíveis na Internet, como por exemplo a consulta do email.

Hoje em dia estes serviços são algo elementar nos dispositivos móveis, que cada vez mais se assemelham a computadores nas funcionalidades que disponibilizam – os *smartphones*. A mudança deve-se à sua capacidade de executar um extenso universo de aplicações e ao facto de permitirem que o utilizador instale novas aplicações complementando o sistema base e moldando-se às suas necessidades e interesses. As suas capacidades de ligação à Internet tanto por Wi-Fi como através das redes móveis celulares permite que os utilizadores possam usufruir dos seus serviços favoritos em qualquer lugar.

Com estes dispositivos surge um novo mercado e novos fluxos de receita através de modelos de negócio baseados na comercialização das aplicações, a maioria desenvolvida por partes independentes ao fabricante do dispositivo. Os dispositivos evoluíram em *Hardware* para permitir a execução de aplicações complexas, que exigem elevadas capacidades gráficas, de processamento e de memória, como por exemplo a execução de jogos com qualidade gráfica elevada e aplicações de navegação. Existem no mercado *smartphones* com capacidades gráficas e de processamento equivalentes a computadores comercializados à 6/7 anos atrás. Estão equipados com ecrãs com qualidade de imagem impressionantes. Os utilizadores procuram desta forma num *smartphone* características muito semelhantes às que já procuravam num computador, que fazem dos *smartphones* dispositivos poderosos tanto para lazer como para trabalho.

As novas funcionalidades e capacidade dos *smartphones* têm conduzido a uma queda de 1.5% ano após ano, na venda dos restantes dispositivos móveis de telecomunicações, segundo

a *International Data Corporation* (IDC)¹ [IDC12e], foram vendidas 398.4 milhões de unidades no primeiro trimestre de 2012, enquanto que no mesmo período de 2011, tinham sido vendidas 404.3 milhões de unidades. Segundo a mesma empresa, nos mesmos períodos, a comercialização de *smartphones* têm vindo a crescer 42.5%, ano após ano.

A escolha de um *smartphone* pode ser bastante complexa, os dispositivos variam tanto nas características como na gama de preços, dependendo do utilizador e das funcionalidades que este espera obter do dispositivo os parâmetros de eleição variam. O mercado está dividido e com a elevada concorrência entre os principais fabricantes, as características de *hardware* equivalem-se, existindo modelos com características e preços semelhantes entre os vários fabricantes. Desta forma, muitos utilizadores focam o seu critério de escolha principalmente no sistema operativo do dispositivo e no seu design. O sistema operativo é uma parte fundamental do dispositivo, este dita não só as aplicações que estão disponíveis para o utilizador mas também o nível de integração com diversos serviços e produtos, introduzindo o utilizador num verdadeiro ecossistema muito para lá do seu *smartphone*.

1.1 Sistemas Operativos Móveis

A necessidade de criar novos serviços e novas funcionalidades nos *smartphones* levaram a que, não só os dispositivos evoluíssem, mas também os seus sistemas operativos. A competição deixou de ser principalmente entre características de *hardware* e fabricantes, para se travar também entre sistemas operativos.

A evolução tecnológica dos *smartphones* criou novos desafios para os sistemas operativos móveis. O sistema operativo é determinante na performance, segurança, e funcionalidades de um *smartphone*[LY09]. Em 2003 o mercado dos *smartphones* era liderado pela Microsoft, Palm OS, Symbian OS, e Linux[LY09]. Atualmente, o Android da Google, o iOS da Apple, e o Symbian da Nokia, são os três sistemas operativos para *smartphones* com maiores cotas de mercado[IDC12a].

■ Android

O Android é um sistema operativo baseado em Linux desenhado principalmente para dispositivos móveis, como *smartphones* e *tablets*, que utilizam processadores ARM. Sendo um Sistema Operativo (SO) que consome poucos recursos, é também incorporado em equipamentos de rede, sistemas inteligentes de televisão, como por exemplo a Google TV, entre outros².

¹IDC - Empresa de pesquisa e análise de mercado, especializada em tecnologia da informação e telecomunicações.

A Google apoiou financeiramente a Android Inc. onde o software começou a ser desenvolvido, mais tarde, em 2005 adquiriu a empresa. A distribuição do Android começou em 2007, sendo anunciada juntamente com a fundação da Open Handset Alliance liderada pela Google[ali10], da qual fazem parte oitenta e seis empresas de hardware, software e de telecomunicações. O Android *Open-Source Project* é responsável pela manutenção e desenvolvimento do sistema operativo.

O Android, foi criado com o objetivo de permitir aos *developers* criarem aplicações móveis, que tirem o máximo partido dos recursos que os *smartphones* têm para oferecer. É construído sobre o *Kernel* do Linux *open-source* e utiliza uma máquina virtual personalizada, que foi desenhada para otimizar os recursos de memória e hardware no ambiente móvel. Por ser código aberto, o Android pode ser livremente expandido de modo a incorporar novas tecnologias que vão surgindo. A plataforma evolui também através da comunidade de programadores, que trabalha em conjunto para construir aplicações móveis inovadoras[ali12].

O desenvolvimento das aplicações para Android é baseado principalmente numa versão personalizada da linguagem Java[Sha07], ou em C/C++, através da utilização do *Native Development Kit*. O *Software Development Kit* (SDK) do Android incorpora um conjunto abrangente de ferramentas de desenvolvimento, que incluem bibliotecas, um depurador, um emulador com base em QEMU, documentação, código de exemplo e tutoriais. Estão também disponíveis ambientes de desenvolvimento (IDEs) para Linux, Mac OS X 10.5.8 ou superior, e Windows XP ou superior. O IDE suportado oficialmente é o Eclipse, utilizando o *Plugin Android Development Tools* (ADT).

As aplicações desenvolvidas são disponibilizadas aos utilizadores através da loja de aplicações online Google Play³, ou através de outros locais, como os sites dos *developers*. A disponibilização de aplicações no Google Play requer registo e o pagamento de uma taxa única de \$25. Existem mais de 600 mil aplicações disponíveis para *download* no Google Play, e uma estimativa de 20 mil milhões de aplicações descarregadas até Junho de 2012, ou uma média de 1.5 mil milhões por mês[Smi12].

O Android tornou-se a plataforma líder mundial para *smartphones* no último trimestre de 2010[Can11]. A meio de 2012, já tinham sido ativados 400 milhões de dispositivos, uma média de 1 milhão de ativações por dia. A versão atual do sistema, é o Android 4.1 Jelly Bean, lançada em Julho de 2012.

²O Android é incorporado até em frigoríficos.

³Introduzido em Março de 2012 quando a Google juntou os serviços de distribuição Google Market (Aplicações) com a Google Music.

■ iOS

O iOS é um sistema operativo proprietário e de código fechado, desenvolvido pela Apple Inc. principalmente para dispositivos móveis. Foi apresentado e lançado em 2007 juntamente com o primeiro iPhone sendo, desde então o sistema operativo em execução nos dispositivos móveis criados pela Apple, como o iPhone, iPad, iPod Touch e também na Apple TV. A sua instalação em dispositivos que não sejam desenvolvidos pela Apple não é permitida.

No lançamento do primeiro iPhone, a Apple não distinguiu o sistema operativo, referindo apenas que o iPhone executava OSX⁴[App08]. Quando o iPhone foi lançado, não suportava aplicações desenvolvidas por terceiros, mas a Apple, ainda em 2007, acabaria por anunciar um kit de desenvolvimento nativo de *software* (SDK), e em Março de 2008 lançou a primeira versão *beta*. Na mesma altura nomeou o sistema operativos de iPhone OS.

Além do iPhone, a Apple lançou também o iPod Touch, com a maioria das capacidades do iPhone exceto as de telefone, e em Janeiro de 2010 anunciou o iPad, que alguns definem como o primeiro *Tablet* bem sucedido no mercado⁵.

Em Junho de 2012, a App Store da Apple tinha disponíveis mais de 650 mil aplicações para iOS e um total de mais de 30 mil milhões de *downloads*[App12d]. Na segunda metade de 2012, tinham já sido ativados 410 milhões de dispositivos[IMA12].

O sistema iOS têm por base o sistema Darwin⁶, sendo por isso um sistema operativo Unix, e responde à tecnologia *multi-touch* de forma fluída. No iOS existem quatro camadas de abstração: a camada *Core OS*, a camada *Core Services* dos Serviços, a camada de *Media*, e a camada *Cocoa Touch*[ios12].

Até 2010 o *multitask* do sistema iOS estava limitado, apenas permitia ao utilizador abrir uma aplicação de cada vez, sendo alterado oficialmente para suportar várias aplicações em simultâneo na versão 4. A versão atual disponível é o iOS 6 lançada oficialmente em Setembro de 2012 juntamente com o iPhone 5[App12f].

O XCode é desde da versão 3.1 o IDE de desenvolvimento para iOS, estando disponível gratuitamente desde Julho de 2010 para todos os utilizadores do sistema operativo OSX Lion e OSX Mountain Lion.

Juntamente com o SDK do iOS, o XCode permite aos *developers* criarem aplicações para iPhone, iPod Touch e iPad, sendo escritas na linguagem Objective-C. O ambiente de desen-

⁴O OS X é um sistema operativo baseado no Kernel Unix OpenDarwin, desenvolvido, fabricado e vendido pela Apple Inc.

⁵A Apple vendeu 11.8 milhões de iPads no primeiro trimestre de 2012, o que corresponde a 68% das vendas totais de *tablets* no mesmo período.[IDC12c]

⁶Em Junho de 2010, a Apple renomeou o iPhone OS para iOS.

⁶Darwin é um sistema operativo livre baseado em UNIX BSD lançado pela Apple Inc. em 2000.

volvimento está equipado com ferramentas de *debug* e análise de performance (*Instruments*), com o *Interface Builder* que permite a criação de interfaces gráficas de forma simples (*drag and drop*), e com um simulador de iPhone e iPad chamado *iOS Simulator*. São também disponibilizados exemplos de código, bibliotecas com artigos e guias, assim como vídeos e documentos explicativos sobre as ferramentas, *frameworks*, métodos de design e boas práticas de desenvolvimento[App12e]. Para que o *developer* possa testar a sua aplicação em dispositivos reais é necessário que esteja inserido no programa de desenvolvimento para iOS da Apple, tendo um custo anual associado de \$99.00.

Apesar do XCode ser gratuito para alguns utilizadores como referido anteriormente, a disponibilização de aplicações fora da App Store (loja oficial de aplicações da Apple) não é oficialmente possível. Os utilizadores apenas podem instalar aplicações através da App Store e por isso, para publicarem as suas aplicações na App Store, os *developers* necessitam de pagar a taxa anual do *Developer Program*, não sendo por isso o desenvolvimento e publicação de aplicações completamente livre.

As aplicações submetidas pelos *developers* para ficarem disponíveis na App Store, passam por um processo de aprovação. Segundo a Apple[App12b], o processo de aprovação de aplicações pretende garantir que as mesmas são de confiança, que têm o desempenho esperado e que não contêm conteúdos explícitos e ofensivos. A aprovação de aplicações, segundo as *guidelines*[App12a] disponibilizadas pela Apple para os *developers*, está dependente do cumprimento de várias regras, distribuídas por 22 secções, sendo rejeitadas aplicações com erros, que não desempenham a função publicitada, que utilizem APIs que não são públicas, entre outras.

Em relação à venda das aplicações, os *developers* podem definir qualquer preço acima de um mínimo fixado, ou optar por as disponibilizarem de forma gratuita. A Apple fica com 30% da receita e os restantes 70% vão para o *developer*[App12c].

■ Symbian

O sistema Symbian foi originalmente desenvolvido pela Symbian Ltd, fundada em 1998 pela Ericsson, Nokia, Motorola e Psion[LY09]. Em 2008 a Nokia adquiriu a totalidade da Symbian Ltd. e foi criada uma organização independente sem fins lucrativos chamada Symbian Foundation, oficialmente lançada em Abril de 2009. A plataforma Symbian foi oficialmente disponibilizada com código aberto em Fevereiro de 2010[Men10].

A Nokia Developer[Nok12] afirma que o Symbian continua a ser uma parte central do seu portefólio durante a transição para o Windows Phone. Além disso, a Nokia anunciou que o Qt é uma pedra fundamental na sua estratégia, referindo que com o Qt como principal ferramenta de

desenvolvimento de aplicações para Symbian, o desenvolvimento para aplicações para o sistema tem futuro.

O sistema Symbian é utilizado numa variada gama de *smartphones*, podendo ser também utilizado em *smartphones* e tablets *touchscreen* a preços relativamente baixos.

Para a criação de aplicações, é utilizado o Qt e o Qt SDK. Quando o Qt não dá acesso a funcionalidades de baixo nível, podem ser utilizadas as *Application Programming Interface* (APIs) Symbian C++ dentro da aplicação Qt. A utilização de Symbian C++, por si só, continua a ser uma opção de desenvolvimento, assim como a linguagem Java, Web Runtime (WRT) e Adobe Flash Lite.

Os criadores de conteúdos têm um suporte abrangente para áudio, imagem e vídeo. Além disso, o Adobe Flash Lite e o *Scalable Vector Graphics Tiny* (SVGT) podem ser utilizados para conteúdos animados, enquanto no *browser* da Nokia para Symbian foram incluídas funcionalidades HTML5.

Os utilizadores de Symbian podem obter as aplicações através da loja *online* OVI⁷ da Nokia. Existe controlo das aplicações por parte da Symbian Foundation, principalmente nas aplicações que necessitam de aceder a ficheiros protegidos do sistema. A versão atual do sistema Symbian é o Nokia Belle (Symbian OS 10.1) lançado em 2011[Mol11].

■ BlackBerry OS

O BlackBerry OS é um sistema operativo utilizado nos dispositivos móveis BlackBerry desenvolvidos pela empresa canadiana RIM. É um sistema operativo proprietário de código fechado, sendo a versão 7.1 a utilizada nos modelos mais recentes. É sistema muito conhecido pelo seu suporte nativo para o mercado empresarial, através do suporte de MIDP 1.0 e mais recentemente de um subconjunto do MIDP 2.0, que permite a ativação completa sem fios e a sincronização do email, calendário, tarefas, notas e contactos, com o Microsoft Exchange, Lotus Domino ou Novell GroupWise quando utilizado através do BlackBerry *Enterprise Server*.

O BlackBerry OS permite a instalação de aplicações desenvolvidas em Java, utilizando *sandboxing* para isolar as aplicações em *runtime* através da Java *Virtual Machine* (JVM), baseada na implementação da Sun do J2ME sendo em parte escrita em C, C++ e *Assembler*.

Em 2009 foi anunciada a loja *online* BlackBerry *App World*, à qual os utilizadores dos novos modelos BlackBerry podem aceder a aplicações aprovadas pela RIM. Na *App World* estão disponíveis algumas aplicações, mas muito distante do número de aplicações disponíveis

⁷OVI Store: <http://store.ovi.com/>

para iPhone ou para Android. As aplicações são tradicionalmente para acesso a *web services* existentes, e menos focadas na experiência para o utilizador[HST10].

Os *developers* podem optar por disponibilizar as suas aplicações através de outros servidores. O desenvolvimento das aplicações é feito utilizando as APIs disponíveis, e as aplicações que utilizam certas funcionalidades restritas, devem ser assinadas digitalmente.

A principal receita da RIM advém da venda dos dispositivos BlackBerry e da implantação de soluções empresariais e de serviços de email. O sistema operativo é apenas uma ferramenta de apoio da RIM para o seu negócio principal, e portanto o desenvolvimento de software por terceiros não é uma prioridade[LY09].

É esperado o lançamento do novo sistema operativo BlackBerry 10 OS em 2013.

■ Windows Phone 7

O Windows Phone 7 é um sistema operativo de código fechado e proprietário desenvolvido pela Microsoft, sendo o sucessor da plataforma Windows Mobile, no entanto são incompatíveis.

Através do lançamento do WP7, a Microsoft passou a direccionar o foco da sua plataforma móvel do mercado empresarial para o mercado do consumidor geral. O WP7 utiliza no seu interface o novo design da Microsoft para os seus sistemas operativos designado por Metro.

O WP7 foi oficialmente apresentado pela Microsoft em 2010, anunciando no mesmo ano o lançamento de 10 dispositivos com o sistema, dos fabricantes HTC, Dell, Samsung, e LG. Em 2011 a Nokia e a Microsoft tornaram-se parceiros, e o WP7 passou a ser o sistema operativo principal dos *smartphones* Nokia, de forma a competir com os ecossistemas Android e iOS.

O desenvolvimento para WP7 é possível através de um conjunto de ferramentas gratuitas disponibilizadas como extensões para o IDE Visual Studio 2010, permitindo o desenvolvimento desde da codificação, à criação da interface gráfica e à depuração. As ferramentas de desenvolvimento estão disponíveis apenas para sistemas operativos Windows Vista SP2 ou superior. A Microsoft oferece ainda gratuitamente a ferramenta Expression Blend para a criação de interfaces do WP7.

O desenvolvimento pode ser feito com recurso a uma versão específica da plataforma Silverlight, que permite a utilização de C# ou Visual Basic, ou à *framework* XNA para desenvolvimento de jogos.

As aplicações podem ser disponibilizadas na Windows Phone Store, loja oficial da Microsoft que em Junho de 2012 tinha mais 100 mil aplicações disponíveis[Bla12](longe ainda dos números das disponíveis para Android e iOS). O acesso à loja oficial é também possível através do Zune Marketplace. A submissão das aplicações está sujeita a certificação por parte da Microsoft.

Em Junho deste ano, a Microsoft anunciou o Windows Phone 8, que será lançado ainda em 2012. O WP8 irá permitir, entre outras coisas, dispositivos com ecrãs maiores, suporte para NFC (para partilha de conteúdos e para realizar pagamentos), a integração do Nokia *Maps*, e a integração de aplicações VoIP. Irá incluir ainda novas ferramentas direcionadas para o mercado empresarial, e a possibilidade de criação de mercados privados para distribuição de aplicações para funcionários. Incluirá ainda suporte para atualizações sem fios.

O WP8 não será disponibilizado como atualização do WP7 para os atuais dispositivos. Contudo deverá ser disponibilizada uma atualização do WP7. Os parceiros para o lançamento do WP8 incluem a Huawei, Nokia, HTC, e Samsung.

1.2 Estatísticas e Cotas de Mercado

Recentes estatísticas das vendas mundiais de *Smartphones* feitas pela IDC [IDC12e], apontam para valores na ordem dos 144.9 milhões de unidades vendidas no primeiro trimestre de 2012. As vendas totais de *smartphones* no ano de 2011 foram estimadas em 491.4 milhões de unidades.

Para o mesmo período de 2012, a Gartner[Gar12] estima que foram vendidas 144.4 milhões de unidades, com uma estimativa de 472 milhões *smartphones*, representando 31% do total de dispositivos de telecomunicações em 2011. Comparando estes números com os números estimados pela mesma empresa (Gartner) em 2010 que foram de 297 milhões de unidades vendidas, representando 19% do total de 1.6 mil milhões de dispositivos móveis vendidos em 2010, a venda de *smartphones* cresceu de 2010 para 2011 59%.

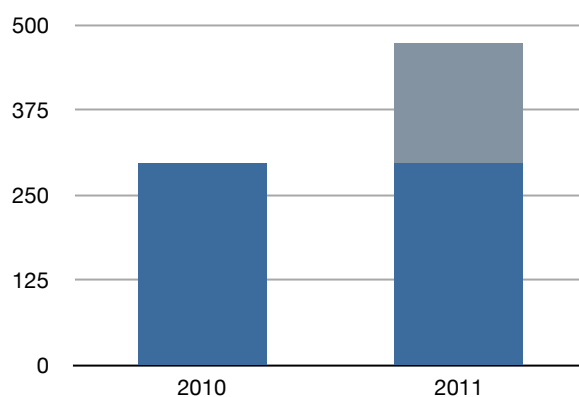


Figura 3.1: Total de *smartphones* vendidos em 2010 e 2011 (Milhões de Unidades)

Na Europa Ocidental, a IDC afirma que foram vendidos 28.8 milhões de *smartphones* no primeiro trimestre de 2011. A comScore[com12] combina os utilizadores de *smartphones* da França, Alemanha, Itália, Espanha e Reino Unido em 44% dos utilizadores de dispositivos móveis em Dezembro de 2011, o que se traduz em 104 milhões de utilizadores.

Segundo um artigo publicado no The Financial Times[The11], uma previsão de JPMorgan aponta para 657 milhões de *smartphones* vendidos em todo o mundo no ano de 2012. A IDC faz uma estimativa[IDC12b] de que a vendas atingirão os 686 milhões de dispositivos no mesmo período. E prevê[IDC11] que irá chegar aos 982 milhões em 2015.

A IMS Research espera que a venda de *smartphones* chegue a 1 bilhão de dispositivos em 2016, o que equivalerá metade do mercado de dispositivos de telecomunicações móveis.

1.2.1 Estatísticas por Fabricante

Segundo as estimativas da IDC[IDC12e] para as quotas de mercado no primeiro trimestre de 2012, a Samsung e a Apple detêm as primeiras posições, com 29.1% e 24.2% respetivamente.

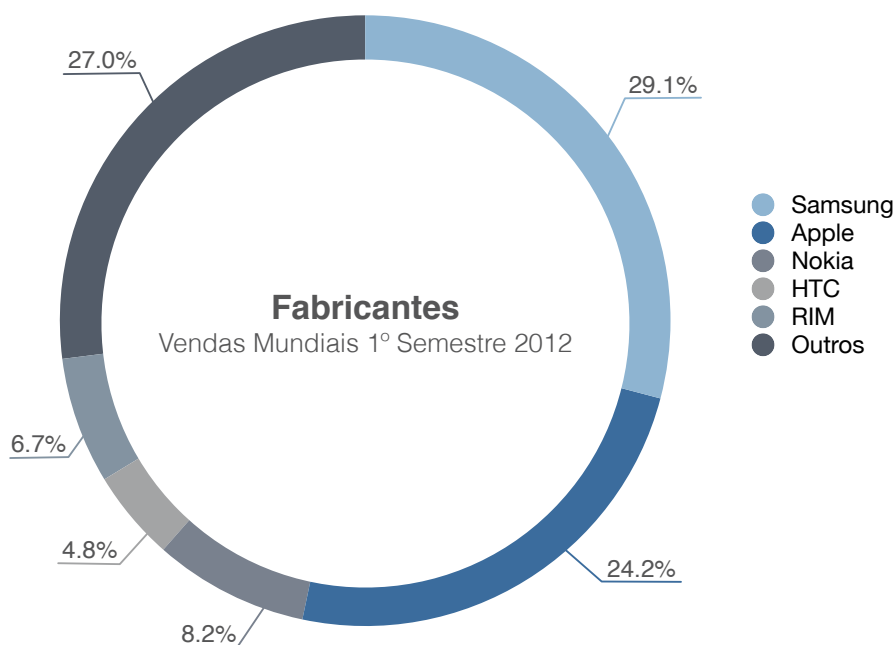


Figura 3.2: Vendas Mundiais de *smartphones* 1º Trimestre de 2012 por fabricante [IDC12e]

A Samsung recuperou a posição de liderança no mercado dos *smartphones* e estabeleceu um novo recorde de mercado do número de *smartphones* vendidos num único trimestre.

A Apple caiu para o segundo lugar no mercado de *smartphones*, mas mesmo assim apresentou um crescimento forte ano-sobre-ano a chegar a 35,1 milhões de unidades vendidas. Os ganhos de mercado da Apple beneficiaram da disponibilização do iPhone em mais operadoras de telecomunicações móveis em todo o mundo, bem como da procura do utilizador final.

As vendas de telemóveis Nokia Symbian caiu vertiginosamente no último trimestre, a procura caiu nos principais mercados emergentes, como por exemplo a China. A empresa optou por fazer uma rápida transição para produtos equipados com o sistema operativo Windows Phone, sobre o qual ele aposta seu futuro no mercado dos *smartphones*.

A Research In Motion, BlackBerry, sofreu um declínio continuado no último trimestre, chegando a níveis não vistos desde 2009. Como a Nokia, a *Research In Motion Limited* (RIM) é uma empresa em transição. *Smartphones* correndo uma nova plataforma, o BlackBerry 10, serão lançados ainda este ano.

No gráfico da figura 3.3 pode observar-se, em termos percentuais, as alterações ano-sobre-ano das cotas de mercado.

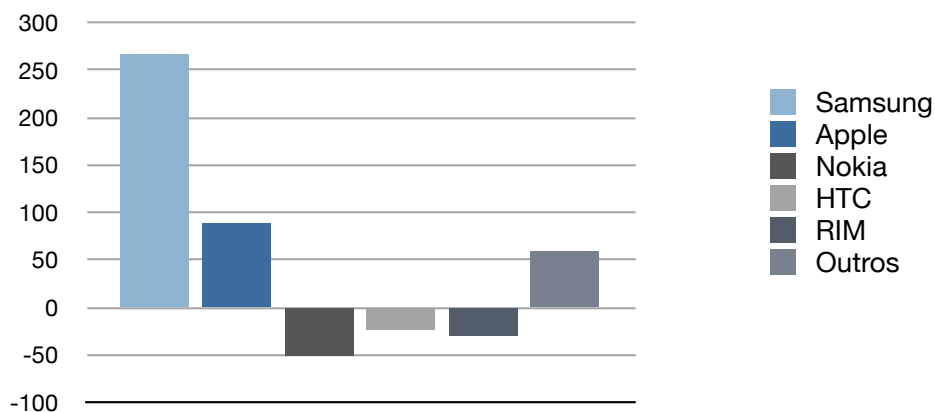


Figura 3.3: Alterações na cota do mercado ano-após-ano (Porcentagem) [IDC12e]

1.2.2 Estatísticas por Sistema Operativo

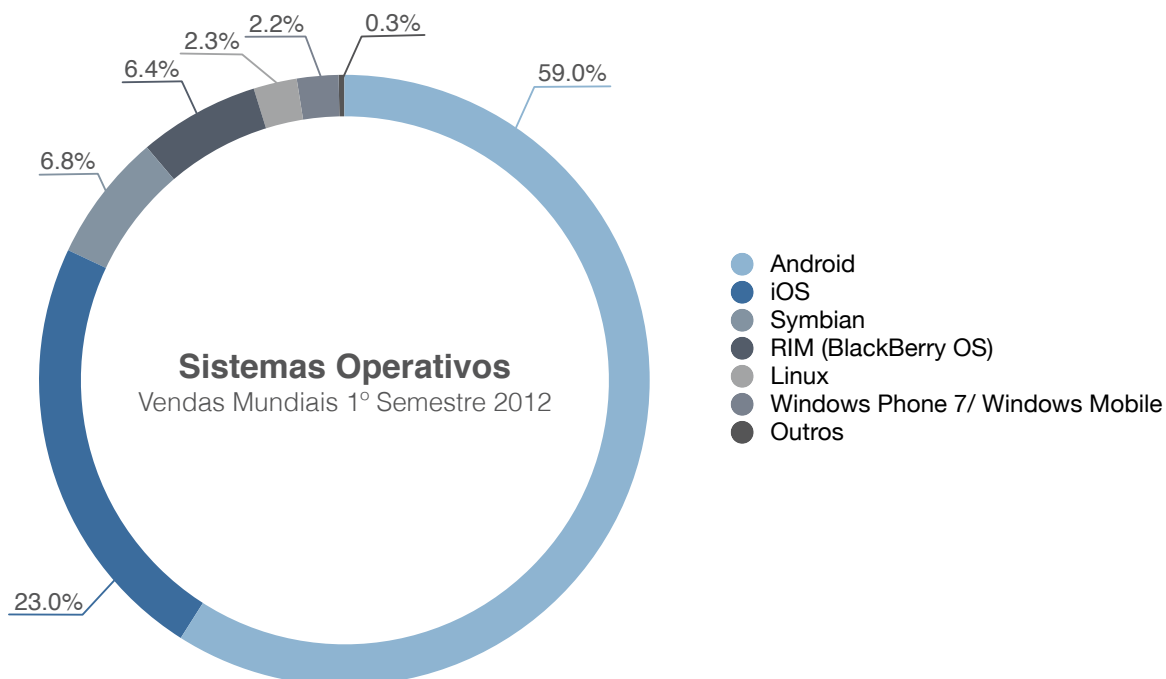


Figura 3.4: Vendas Mundiais de *smartphones* 1º Trimestre de 2012 por SO [IDC12a]

O Android terminou o primeiro trimestre de 2012 como líder global entre os sistemas operativos móveis, correndo em mais da metade de todos os *smartphones* vendidos. Além disso, o Android impulsionou uma longa lista de parceiros fornecedores de *smartphones*. A Samsung

foi o maior contribuinte para o sucesso do Android, respondendo por 45,4% de todas as vendas de *smartphones* baseados em Android. Mas, além da Samsung, várias outras empresas viram crescer os seus volumes de vendas.

O iOS da Apple registou um crescimento ano-sobre-ano forte, graças à procura sustentada do iPhone 4S, e a adição de inúmeras operadoras de telecomunicações móveis que disponibilizaram o iPhone pela primeira vez.

O sistema Symbian teve o maior declínio ano-sobre-ano, um resultado impulsionado pela transição da Nokia para Windows Phone. Mas, mesmo com a diminuição dos volumes de vendas de *smartphones* com Symbian, continua a existir procura do sistema operativo. Além disso, a Nokia continua a suportar o mesmo. Ainda assim, como a Nokia enfatiza o sistema Windows Phone, a IDC espera novas quedas do Symbian no decorrer deste ano.

A BlackBerry continuou a sua trajetória de queda devido à quebra na procura de dispositivos BlackBerry antigos. O mercado aguarda o lançamento oficial do BlackBerry 10 e de *smartphones* baseados no mesmo ainda este ano. Além disso, muitas empresas já permitem que os utilizadores levem os seus próprios *smartphones*, permitindo que os sistemas operativos concorrentes tirem quota de mercado ao BlackBerry. Embora a RIM não tenha oficialmente lançado o BlackBerry 10, vislumbres iniciais da plataforma têm mostrado melhorias.

O Linux manteve sua pequena presença no mercado de *smartphones* em todo o mundo, em grande parte graças à ênfase contínua da Samsung em Bada. Até o final do trimestre, a Samsung representava 81,6% de todos os *smartphones* com Linux, um ganho de 3,6% em relação ao mesmo período do ano anterior. Outros fornecedores, entretanto, têm feito experiências com Android para impulsionar o volume de vendas. Ainda assim, a sorte do Linux está muito dependente da estratégia da Samsung, que já engloba Android, Windows Phone, e ainda este ano, Tizen.[IDC12a]

2 Principais Tecnologias Disponíveis nos *Smartphones*

Muitos serviços e aplicações apenas são utilizáveis ou úteis, com acesso à Internet; os *smartphones* atualmente são dispositivos desenhados para funcionarem com conectividade constante. Com a evolução dos *smartphones* e dos modelos de negócio baseados em lojas online de aplicações, assim como com a introdução dos primeiros serviços de armazenamento na *Cloud*⁸, tornou-se indispensável a ligação à Internet para tirar total partido do dispositivo.

Praticamente todos os *smartphones* estão equipados com tecnologia para acesso a redes Wi-Fi, contudo não é possível ligação constante em todo lado através de Wi-Fi, sendo por isso cada vez mais utilizado o serviço de dados das redes móveis celulares. Por este motivo, as redes móveis celulares tem evoluído no sentido de suportar cada vez mais dispositivos e taxas de transferência mais elevadas, de modo a suportarem os serviços disponíveis nos *smartphones*, os quais exigem maior largura de banda, como por exemplo o *streaming* de vídeo e televisão.

O recente aparecimento da tecnologia LTE que aparece como tecnologia de quarta geração, é mais um passo neste sentido, permitindo aos dispositivos móveis taxas elevadas de transmissão de dados, e oferecendo uma qualidade de serviço superior através de uma melhor cobertura. Ainda não existem no entanto muitos dispositivos no mercado com tecnologia LTE. A maioria dos dispositivos funciona através da rede móvel 2G ou 3G.

Por outro lado, o Bluetooth já há muito tempo que está disponível em praticamente todos os dispositivos móveis, permite a transferência de dados entre dispositivos e a ligação a periféricos.

O GPS tornou-se uma tecnologia indispensável nos *smartphones*, independentemente da gama de preços, sendo utilizado atualmente não apenas para navegação, mas também para serviços baseados na posição do utilizador.

Todas estas tecnologias podem ser utilizadas para analisar o movimentos dos utilizadores. A utilização do GPS é comum neste tipo de tarefas, no entanto, é também possível analisar o movimento dos utilizadores num determinado espaço através da recolha das informações sobre o ambiente rádio envolvente, por exemplo, através dos *beacons* emitidos pelos APs das redes Wi-Fi e do nível de sinal recebido, ou utilizando as informações das células GSM detetadas. Através do Bluetooth é possível detetar relações de proximidade num determinado instante.

Uma explicação detalhada sobre cada uma destas tecnologias encontra-se disponível no apêndice A.

⁸ *Cloud* - Modelo de armazenamento *online* onde os dados são armazenados em servidores que são geralmente de entidades terceiras.

Capítulo 4

Estudo do Problema

1 Conteúdos para o Utilizador

Conceber uma solução de *sensing*, que transforma os dispositivos móveis pessoais, em sensores móveis que permitam observar o movimento humano no espaço físico, corresponde a dar resposta a um conjunto de desafios não só tecnológicos, mas também sociais. Os conteúdos ou serviços da aplicação que se pretende desenvolver, surgem como resposta aos seguintes desafios:

- Motivar os utilizadores a utilizarem a aplicação.
- Maximizar o “tempo de vida da aplicação”. Especifica-se como tempo de vida da aplicação, o período desde que o utilizador instala e começa a utilizar a aplicação, até que o utilizador deixa de utilizar a mesma.
- Maximizar o número de utilizadores.
- Maximizar o tempo de interação com a aplicação.
- Maximizar o tempo de conexão à rede.
- Lidar com questões de privacidade dos utilizadores.
- Definir um modelo de negócio claro para os utilizadores.

1.1 Análise dos Modelos de Negócio

A definição de um modelo de negócio, deve preceder o desenvolvimento de uma aplicação ou de um serviço. O modelo de negócio deve ser definido para salvaguardar não só os interesses da aplicação, mas também os dos utilizadores. É importante encontrar o equilíbrio para que a aplicação tenha sucesso.

No âmbito dos objetivos desta dissertação, o modelo de negócio representa como os utilizadores são recompensados pela cedência dos seus dados de movimento. Como se pretende o desenvolvimento de uma aplicação, os modelos de negócio abordados anteriormente, no estado da arte, constituem um ponto de partida para a especificação de um modelo que satisfaça o propósito deste projeto. Assim sendo, o primeiro passo é definir se o modelo de negócio será baseado na aplicação como produto ou como serviço. Para isso, devem ser consideradas as vantagens e desvantagens de cada abordagem em relação aos objetivos.

Através dos modelos de negócio genéricos de software, como produto e como serviço, definiu-se um modelo de negócio específico para este projeto (Fig.4.1). Na figura estão representadas três entidades: o utilizador, o projeto SUM, no qual esta dissertação se enquadra, e outros projetos, por exemplo de outras universidades com os quais poderá futuramente existir a partilha dos dados.

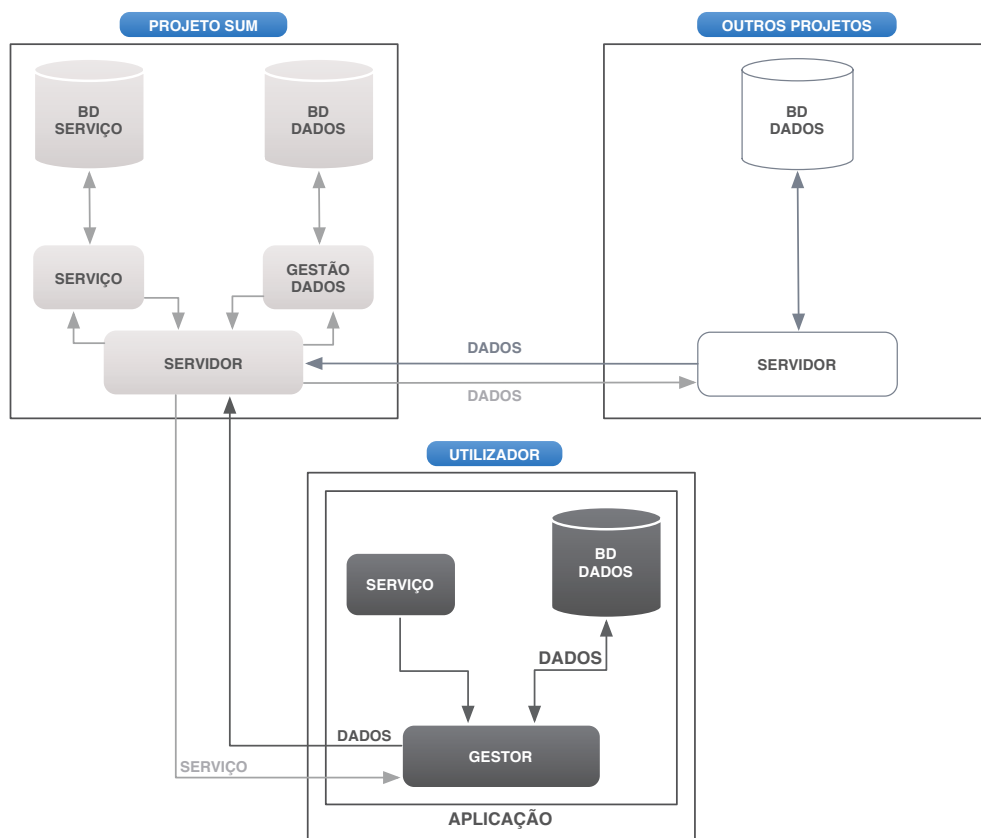


Figura 4.1: Modelo de negócio geral

O modelo anterior (Fig. 4.1), engloba as duas opções consideradas para este projeto, um modelo baseado na aplicação como produto ou como serviço (Fig. 4.2(a) e Fig. 4.2(b)). No caso do modelo da aplicação como produto (Fig. 4.2(a)), estão disponíveis funcionalidades integradas na aplicação, que funcionam como recompensa para o utilizador. Neste modelo, apenas existe interação com o servidor, para o envio dos dados de movimento. No caso da aplicação ser disponibilizada como serviço (Fig. 4.2(b)), a recompensa é baseada numa aplicação que permite ao utilizador o acesso a um serviço, prestado pelo servidor.

As vantagens do modelo baseado na aplicação como serviço em relação à aplicação como produto são consideráveis. No caso da aplicação como produto a utilização da aplicação é independentemente da ligação ao servidor, não sendo esta situação favorável para o envio dos dados de movimento recolhidos. No caso da aplicação como serviço a aplicação disponibiliza conteúdos que estão no servidor, tendo obrigatoriamente de ser estabelecida ligação com o servidor de forma a utilizar as principais funções da aplicação, garantindo uma oportunidade para o envio dos dados.

Considerando estes factos, o modelo de negócio mais indicado para este projeto é o que tem por base a aplicação como serviço, assim sendo este modelo de negócio define a natureza da aplicação e do serviço que a mesma disponibiliza.

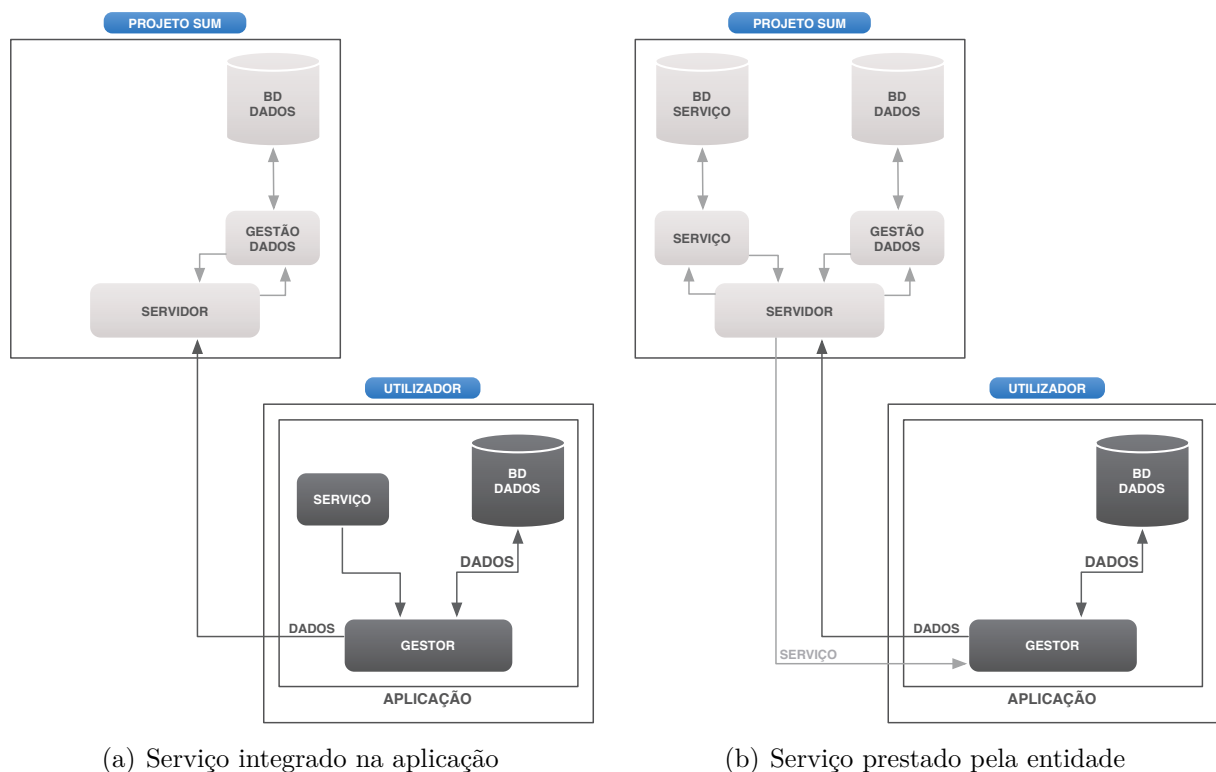


Figura 4.2: Sub-modelos de negócio (a)

1.2 Aplicação Quiz

A primeira aplicação pensada e especificada, consiste num jogo. A motivação para esta aplicação advém do interesse intrínseco das pessoas em aplicações lúdicas e da competitividade humana. O jogo é baseado em perguntas e respostas. As perguntas são enviadas aos utilizadores a determinadas horas do dia. Se, num determinado período, o utilizador se ligar à Internet, receberá a pergunta no seu *smartphone*.

Ao responderem às perguntas, os utilizadores ganham ou perdem pontos, dependendo da resposta dada estar correta ou errada. Os pontos acumulados pelos utilizadores são contabilizados para o *Ranking* geral. As perguntas enquadram-se em diversas categorias e graus de dificuldade, podendo ser de escolha múltipla (A, B, C e D), ou apenas de “Sim” ou “Não”. Dependendo do número de opções, difere no número de ponto descontados no caso da resposta estar errada.

No jogo, o utilizador poderá completar objetivos pessoais e responder a perguntas bónus, que são enviadas aos utilizadores a uma hora aleatória. Apenas os utilizadores ligados no exato momento em que a pergunta bónus é enviada, têm hipótese de responder.

1.2.1 Descrição do Jogo

Se existir ligação à Internet, são recebidas notificação das novas perguntas disponíveis. As notificações informam até que horas a mesma pode ser respondida, o tempo para responder e a categoria da pergunta.

Ao receber estas informações, o utilizador pode responder de imediato à pergunta, ou deixar para responder mais tarde, ficando na área de notificações do jogo, acessível até à hora limite.

■ Categorias

As diferentes categorias de perguntas são distribuídas geograficamente por áreas. Quando os jogadores acumulam um certo número de pontos, podem ser criadas novas áreas de uma determinada categoria, nas coordenadas geográficas em que se encontram. Nestas áreas, a probabilidade de serem recebidas perguntas dessa categoria é mais elevada.

■ Perguntas Bónus

As perguntas bónus são enviadas para os utilizadores a uma hora aleatória e são apenas recebidas pelos jogadores ligados no momento. Estas perguntas valem mais pontos, no entanto uma resposta errada também retira mais pontos. A categoria das perguntas bónus é aleatória e desconhecida.

■ Enigmas

Os enigmas estão disponíveis até serem desvendados, sendo a resposta uma única palavra que o utilizador deverá introduzir. Ao desvendar o enigma, o utilizador tem acesso a uma coordenada geográfica no seu mapa, devendo deslocar-se ao local para colecionar a recompensa. A cada enigma estão associados diferentes pontos geográficos, dependendo da cidade em que se encontra o utilizador, isto para que um utilizador não tenha de se deslocar demasiado. No entanto, quando a recompensa é colecionada numa coordenada, todas as restantes coordenadas relativas a esse enigma ficam inativas.

■ *Ranking*: Jogadores, Estatísticas e Estrelas

São contabilizados os pontos acumulados e conquistas pessoais, para determinar a posição do jogador no *Ranking* Geral. As conquistas pessoais são representadas por estrelas (Bronze, Prata, Ouro e Platina), que o utilizador poderá conquistar respondendo consecutiva e corretamente a um determinado número de perguntas. O número de respostas corretas consecutivas aumenta consoante a estrela que está a ser conquistada. Poderá estar disponível a opção de publicar as estatísticas e conquistas no Facebook.

1.2.2 Conteúdos

A base de dados das perguntas e respostas para o jogo é um ponto crítico da conceção desta aplicação. Emergem desafios associados tanto à criação, como à manutenção da mesma. Foram consideradas as seguintes alternativas:

■ A Geração das Perguntas é feita pelos Administradores em *Back Office*.

A vantagem advém da possibilidade de garantir a fiabilidade e correção das perguntas, assim como o controlo da linguagem e do nível de dificuldade das perguntas. No entanto, a consequente desvantagem é a necessidade de um administrador para as tarefas de *back office*.

■ As Perguntas e Respostas são Introduzidas pelos Utilizadores

Como vantagem, é possível dispensar o trabalho de "back office" e melhorar a interatividade com os utilizadores. Este pode ser um fator de motivação para a utilização. Dando a oportunidade aos utilizadores de participarem na criação e evolução do conteúdo da base de dados torna a aplicação mais interativa, sendo que interação significa tempo de utilização.

No entanto, surge a desvantagem de não se garantir o controlo dos conteúdos, tanto do ponto de vista de linguagem, como da correção das perguntas e respostas. No entanto, esta desvantagem poderá ser minimizada através de mecanismos que permitem aos utilizadores reportarem perguntas incorretas ou inadequadas, de modo a que o conteúdo seja removido do sistema.

Mesmo através da implementação do mecanismo de denúncia, esta situação poderá revelar-se um problema. Se o nível de correção das perguntas for reduzido, poderá conduzir ao desinteresse pela aplicação.

■ Utilização de Bases de Dados Existentes

A utilização de um serviço já existente, que forneça uma API para o acesso a uma base de dados de perguntas e respostas, poderá ser uma solução viável, dependendo da qualidade dos conteúdos. Pode-se avaliar a qualidade da base de dados pelos seguintes parâmetros:

- Quantidade de perguntas disponíveis.
- Período de atualização dos conteúdos.
- Nível de correção dos conteúdos disponíveis.
- Disponibilidade do serviço.
- Simplicidade da API de acesso.
- Custo do serviço.

Esta abordagem, caso cumpra os requisitos anteriores na medida do razoável, permite que o conteúdo seja controlado, não sendo necessário trabalho de *back office*.

No entanto, esta abordagem requer a interação entre o sistema e essa base de dados, não existindo garantias de disponibilidade do serviço, que pode ser encerrado ou ficar indisponível a qualquer altura. Adicionalmente, os idiomas disponíveis são limitados.

1.3 Aplicação GeoAnúncios

As taxas de desemprego atingiram este ano valores históricos, existindo poucas ofertas para um crescente número de desempregados. Além disto, as ofertas disponíveis nem sempre são simples de encontrar, passando por vezes despercebidas. Por estes motivos, concebeu-se uma aplicação com o objetivo primário de ajudar as pessoas desempregadas na complicada tarefa de procura de emprego.

Acresce ao problema do desemprego a crise económica que afeta muitos estabelecimentos e prestadores de serviços. Sectores como a restauração e comércio encerram por falta de clientes que gerem receita para a sua sustentação económica. Alguns dos que conseguem manter a atividade não têm disponibilidade para investir em publicidade e divulgação. De forma a auxiliar também nestes casos, alargou-se o âmbito desta aplicação para a publicidade e divulgação de serviços, comércio, ofertas de aluguer de habitações, entre outras.

O objetivo da publicidade, dos anúncios de emprego e de habitação é o mesmo, abranger o maior número de pessoas possível. A publicidade através de anúncios nos jornais, da distribuição de panfletos e cartazes e de anúncios na rádio e na televisão, implica custos. Por outro lado, algumas pessoas optam por pessoalmente afixarem anúncios em determinados pontos da cidade ou colocar a publicidade nos próprios estabelecimentos. Porém estes métodos não são tão eficientes, a sua abrangência é limitada pela sua condição estática, sendo informadas apenas as pessoas que passam no local e que se apercebem dos anúncios.

Por outro lado, nem sempre é simples para quem procura encontrar anúncios do seu interesses. A procura de emprego por exemplo, através de anúncios nos jornais, torna-se um processo desgastante e complicado. Promoções interessantes, por vezes passam despercebidas devido ao elevado número de informação que é difundida de forma heterogénea, sendo complexo filtrar o que realmente interessa.

A aplicação pensada surge como reposta a estes desafios, possibilitando a publicação e a consulta de anúncios referenciados geograficamente, que englobam diversas categorias. A aplicação fornece mecanismos e filtros de pesquisa, para auxiliar o utilizador a encontrar de forma simples e eficiente o que procura.

As vantagens desta aplicação são consideráveis em relação aos serviços disponíveis e usuais.

● Vantagens para quem anuncia:

- Facilidade de publicação de anúncios, diretamente através do *smartphone* e disponibilidade instantânea para consulta.
- Georreferência. O local do anúncio é automaticamente definido através da posição atual ou pode ser definido manualmente pelo anunciante.
- É um serviço gratuito.
- Elevada abrangência. Os anúncios não ficam restritos a uma área.

● Vantagens para quem procura:

- Pesquisa rápida e controlada.
- Informação centralizada. Diversos tipos de anúncios numa só aplicação.
- Mobilidade. O utilizador poderá deslocar-se por uma cidade, consultando os anúncios perto da sua posição atual.
- Possibilidade de organização e mapeamento. Os anúncios podem ser visualizados no mapa, um utilizador que procura emprego poderá traçar a sua rota diária, organizando as suas deslocações.
- Facilidade de deslocação ao local do anúncio. Não é necessário tomar nota de moradas ou procurar ruas. Basta selecionar um anúncio e através da aplicação de GPS o trajeto é calculado.

1.3.1 Descrição da Aplicação

■ Créditos

Sendo uma aplicação gratuita, não existe qualquer custo de utilização. No entanto são necessários créditos para a publicação dos anúncios, que vão sendo atribuídos aos utilizadores quando recebidos no servidor os dados recolhidos pelo seu dispositivo. A limitação da publicação de anúncios através dos créditos é parte constituinte do modelo de negócio definido, que será abordado mais à frente na secção 1.5.

■ Publicação de um Anúncio

Cada anúncio deve enquadrar-se numa das categoria padrão, sendo definido um título, e uma breve descrição com uma breve exposição dos detalhes importantes, como por exemplo o contacto; adicionalmente é possível configurar o período de disponibilidade do anúncio. Os anúncios requerem também a definição de uma posição geográfica. Para a publicação ser concluída, é necessária a conectividade à Internet.

■ Pesquisa

A pesquisa confere aos utilizadores uma forma de encontrarem anúncios do seu interesse. Para a obtenção de resultados mais restritos, pode ser escolhida apenas uma categoria, aplicados filtros e efetuarem-se pesquisas por termos textuais.

■ Mapa e Navegação

A posição geográfica dos anúncios torna possível a visualização dos anúncios no mapa e a navegação para os locais dos mesmos. O trajeto efetuado durante a navegação poderá ser recolhido como um tipo especial de dados, contextualizado pela categoria do anúncio. Vejamos um exemplo, os trajetos de utilizadores que visitam locais de anúncios de emprego, tornam possível a análise de padrões de movimento deste grupo populacional.

■ Registo

O registo é um requisitos na maioria dos serviços disponíveis *online*, garantindo o controlo no acesso. Adicionalmente, confere ao administrador a possibilidade de bloquear o acessos e garante que um utilizador apenas pode gerir os seus conteúdos autenticando-se. Por estes motivos, o registo é acompanhado de vantagens para o utilizador e para a administração do sistema.

No caso da aplicação/ serviço GeoAnúncios, é necessário analisar esta questão por outro ponto de vista. A aplicação pretende atrair os utilizadores, funcionando como recompensa pelos dados de movimento recolhidos. Para o sistema, o registo permite associar os conteúdos aos utilizadores e atribuir créditos. No entanto, devido à recolha dos dados de movimento, o registo poderá ter influência negativa na adesão. Mesmo com as medidas que serão implementadas para salvaguardar a privacidade dos dados, psicologicamente para os utilizadores o seu registo está associado ao serviço e portanto aos dados de movimento recolhidos.

Este efeito pode ser minimizado requerendo apenas a informação indispensável no registo. O email e uma palavra-passe é suficiente, para que seja possível:

- O contacto com o utilizador.
- Validação do registo através do email.
- Minimizar o impacto psicológico de perda de privacidade. Os utilizadores podem criar um email apenas para utilizarem o serviço GeoAnúncios.
- Minimizar a complexidade do registo e da gestão da conta.

1.4 Conclusões

Com o objetivo de selecionar uma das duas aplicações especificadas para ser desenvolvida, foi necessário analisar em que medida as funcionalidades de cada uma dão resposta aos desafios de recolha de dados.

1. Pretende-se motivar a utilização da aplicação, maximizando desta forma o número de utilizadores. Dependendo do nível de motivação, poderá variar tanto o tempo de interação com a aplicação, como o tempo de vida da mesma.

■ **Quiz:** Os enigmas, os objetivos pessoais e a competitividade, são elementos motivadores de interesse nos utilizadores.

● **GeoAnúncios:** É uma aplicação que prove à população um serviço gratuito, de elevada abrangência e utilidade.

2. O tempo de conexão à Internet é essencial para o envio dos dados de movimento recolhidos.

■ **Quiz:** As perguntas bónus maximizam o tempo de ligação, visto que somente os utilizadores com ligação à Internet no instante do envio da pergunta a recebem, assim incentiva-se a conectividade regular.

● **GeoAnúncios:** A maioria das funções disponíveis são dependentes da ligação à Internet. A acumulação de créditos necessários à publicação de anúncios, implica o envio dos dados.

3. O modelo de negócio deve ser claro, de forma a que os utilizadores compreendam as vantagens de fornecer os seus dados de movimento.

Em ambas as aplicações os utilizadores serão elucidados da importância da cedência dos seus dados para a investigação académica. Adicionalmente, algumas funcionalidades das aplicações justificam a recolha de dados para o seu funcionamento, por exemplo de GPS.

■ **Quiz:** A criação de áreas, é uma das funcionalidades que requerem a utilização da posição do utilizador, assim como os enigmas, que implicam a deslocação do utilizador a uma dada coordenada.

● **GeoAnúncios:** Para os anunciantes, é inteligível que apesar da aplicação ser grátis, a publicação permanece limitada pelos créditos, evidenciando as vantagens da cedência dos dados. A utilização da posição do utilizador, é justificada em certa medida pelas funcionalidades de visualização de anúncios no mapa e da navegação por GPS.

Após a análise das características das duas aplicações, conclui-se que ambas são boas soluções. No entanto, a aplicação GeoAnúncios oferece melhores perspetivas, podendo vir a ser um serviço público, útil à população em geral e uma ferramenta importante para os desafios que o desemprego coloca.

Do ponto de vista técnico, os requisitos das duas aplicações são similares, sendo que no caso do *Quiz*, não foi possível encontrar uma solução satisfatória para a produção e manutenção dos conteúdos.

A aplicação GeoAnúncios reúne um conjunto abrangente de desafios técnicos e tecnológicos, pois incluirá diversas funcionalidades.

1.5 Modelo de Negócio – GeoAnúncios

Ao longo destes primeiros capítulos foram abordados os modelos de negócio de software. Partindo do conceito mais abrangente e comercial, foi sendo delineado um modelo no qual se enquadram os objetivos desta dissertação. Considerando que este projeto é de natureza académica, será mais correto denominar o modelo final de **modelo de recompensa**.

O modelo final da aplicação e serviço GeoAnúncios é ligeiramente diferente do modelo base adotado anteriormente. Esta diferença deve-se à introdução do conceito de créditos como recompensa adicional. Os créditos firmam a necessidade do envio dos dados, salvaguardando os interesses do projeto.

O modelo de negócio para este serviço está ilustrado na figura 4.3. Foi considerada a possibilidade da inclusão posterior de uma entidade denominada de “Outros Projetos”, apesar de não existir nenhuma parceria com outra entidade atualmente. No entanto, futuramente os dados de movimento recolhidos podem ser relevantes para outros estudos e investigações.

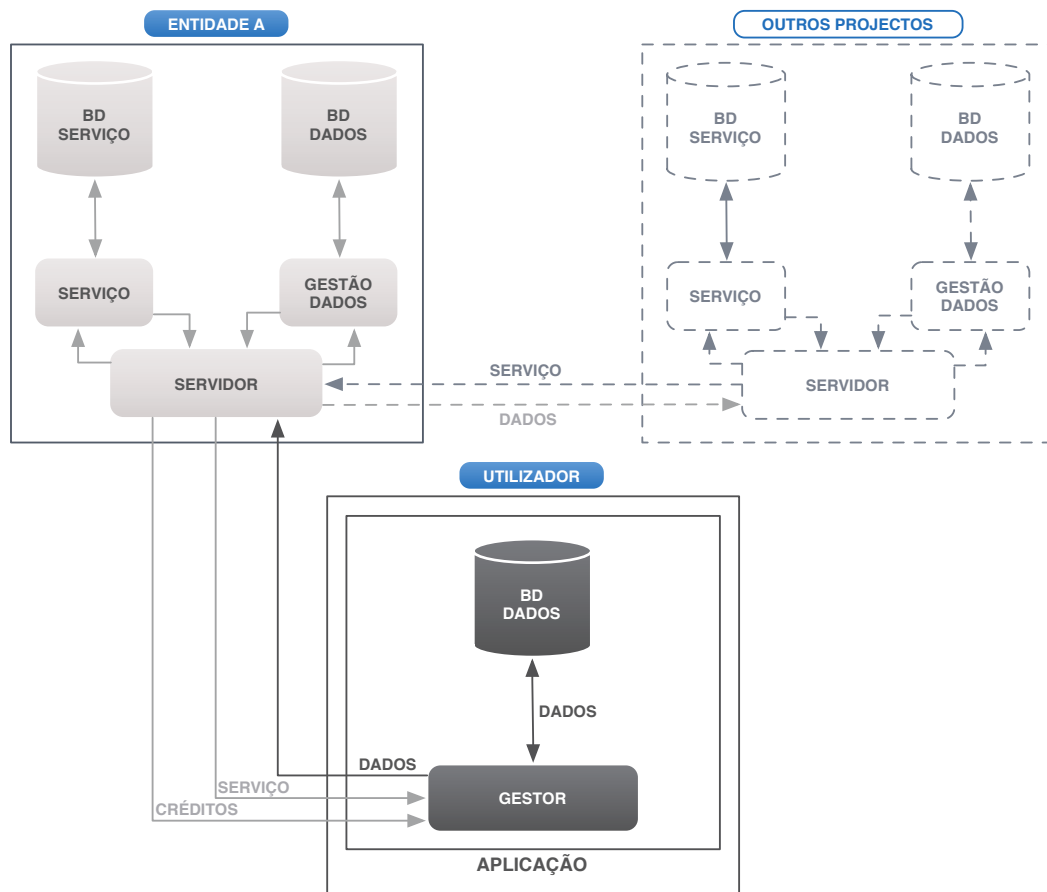


Figura 4.3: Modelo de recompensa do serviço GeoAnúncios

2 Recolha de Dados e Poupança de Energia

Uma das limitações primárias dos dispositivos móveis é a autonomia da bateria, sendo uma característica que tem evoluído pouco ao longo do tempo, quando comparada por exemplo com os ecrãs ou processadores.

A autonomia da bateria varia de acordo com a sua capacidade, com nível de utilização dos recursos do *smartphone* e os seus respetivos consumos energéticos. O gráfico da figura 4.4 apresenta os consumos¹ em mA², da utilização dos principais componentes de um *smartphone*.

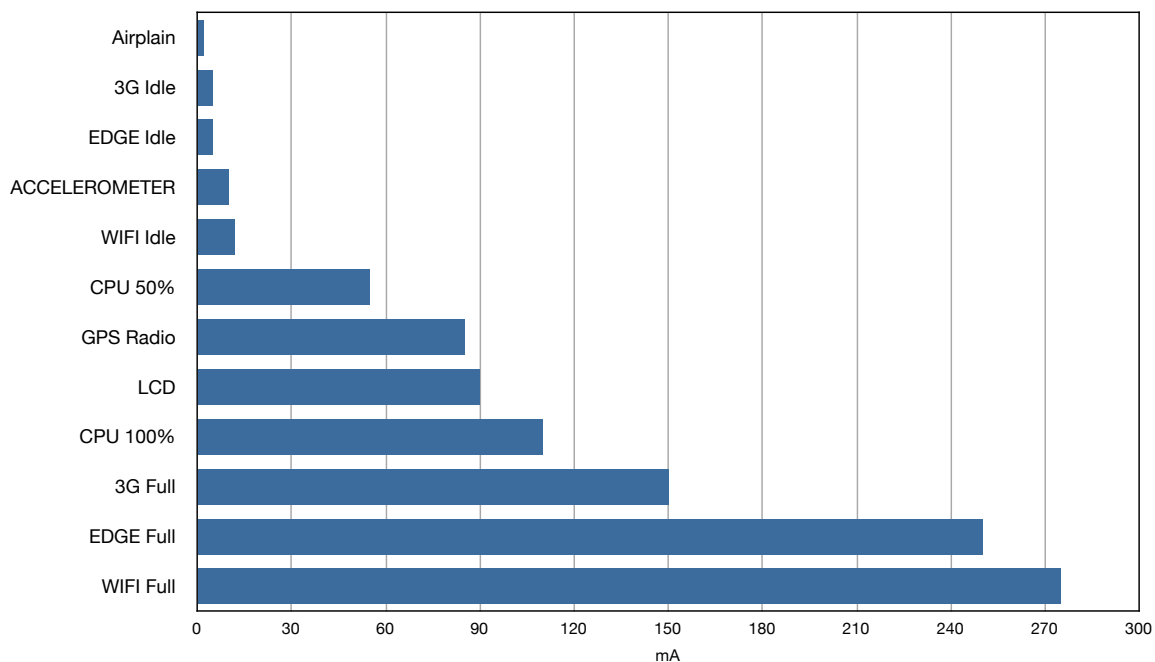


Figura 4.4: Carga elétrica utilizada por unidade de tempo num *smartphone*[Sha09]

É perceptível, através do gráfico, que interfaces como o GPS e o *Wi-Fi* implicam a utilização de valores elevados de carga elétrica, sendo acrescido do consumo do CPU, necessário para a utilização destes interfaces.

É evidente para os utilizadores que o consumo de bateria aumenta consideravelmente quando utilizam o GPS para se orientarem num determinado percurso. Através dos valores apresentados no gráfico, é simples entender o consumo de energia neste caso; acresce aos gastos da utilização do GPS, o consumo do LCD e do CPU.

¹Valores medidos utilizando um monitor de potência industrial com taxa de amostragem 5kHz, sendo utilizada a potência média com menor desvio padrão.

²Ampere é uma medida da quantidade de carga elétrica que passa num ponto de um circuito elétrico por unidade de tempo.

Ainda através do gráfico, é possível constatar que o Wi-Fi, mesmo em modo inativo consome energia. Deste modo, alguns utilizadores optam por desligar interfaces como o *Bluetooth* e o *Wi-Fi* quando não os estão a utilizar para eliminar estes consumos.

Dos interfaces necessários neste projeto, o GPS, o *Bluetooth* e o *Wi-Fi*, são os que representam o impacto mais significativo na autonomia do dispositivo. Na tabela 4.1, são apresentados os valores médios do consumo de cada interface, quando inativos e em utilização.

	Interface			
	Wi-Fi	GPS	Bluetooth	GSM
Estado				
Consumo em inativo	12 mA	0 mA	6 mA	Não considerado
Consumo em utilização	275 mA	85 mA	50 mA	Não considerado
Duração do scan	1.2 seg	11 seg	13 seg	Não considerado
Scan Automático	De 6 em 6 seg	Não	Não	Não considerado

Tabela 4.1: Consumos e duração de um *scan* por interface

Os consumos de *Wi-Fi* e do GPS apresentados na tabela 4.1, foram obtidos na apresentação feita por Jeff Sharkey, engenheiro de software da Google, na conferência Google IO em 2009[Sha09]. Os consumos do *Bluetooth*, foram obtidos através de *datasheets* de módulos de *Bluetooth* v2.1 Classe 2, semelhantes aos utilizados na maioria dos *smartphones*. Não são considerados gastos na utilização da Rede Móvel Celular (GSM), isto porque os *smartphones* no seu funcionamento normal obtêm e mantêm as informações da Rede Móvel Celular necessárias para este projeto.

O tempo de *scan* do *Wi-Fi*, é o tempo mínimo que demora a serem percorridos os 12 canais de frequência, multiplicado por 100 ms do Wi-Fi AP *Broadcast Beacon*.

Em relação ao GPS, segundos testes efetuados, são necessários pelo menos 11 segundos após determinada a posição do utilizador, para serem obtidas coordenadas com precisão entre 2 e 32 m, que são valores satisfatórios. A variação da precisão, depende se o utilizador está dentro ou fora de um edifício. Em certos casos, principalmente dentro de edifícios, é difícil obter-se um valor nesta ordem de valores em 11 segundos, contudo aumentar o período acima de 11 segundos, implica aumentar os consumos energéticos.

No caso do *Bluetooth*, o tempo de *scan* (geralmente referido como tempo de descoberta), é de 10.24 segundos na melhor das hipóteses, segundo as especificações[Blu12]. O tempo de descoberta pode prolongar-se dependendo da quantidade de interferências de rádio, como outros

dispositivos *Bluetooth* e redes Wi-Fi, que utilizam a mesma parte do espectro de radiofrequências, a banda de 2,4 GHz. Testes realizados com dois *smartphones* apontam para uma média de 13 segundos de tempo de descoberta.

2.1 Recolha Contínua

Considerando os consumos energéticos apresentados na tabela 4.1, realizou-se uma estimativa do impacto da recolha de dados não controlada (amostragem contínua dos interfaces (Fig.4.5)) na autonomia da bateria de um *smartphone*.

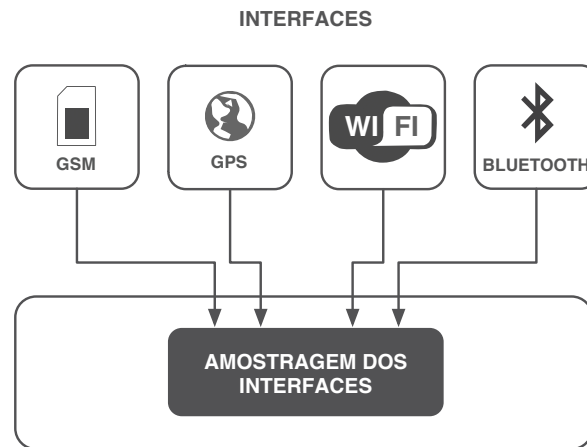


Figura 4.5: Amostragem não controlada dos Interfaces

Depois de realizados os cálculos, obtiveram-se os valores representados no gráfico da figura 4.6, o que reflete o consumo da recolha constante durante 60 minutos, sendo considerada uma bateria com a capacidade de 1200 mAh, comum em *smartphones*. O objetivo é estimar apenas os gastos dos interfaces, desprezando assim os gastos do CPU e considerando que o LCD está desligado durante o processo.

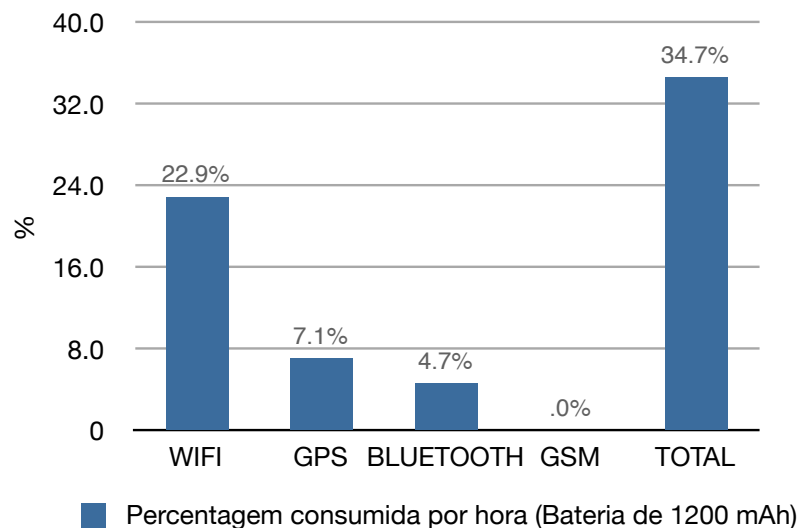


Figura 4.6: Estimativa da percentagem de bateria consumida por hora em recolha contínua

As estimativas anteriores, permitem concluir que uma amostragem contínua consome a bateria (de 1200 mAh) por completo em aproximadamente 3 horas, comprovando que uma amostragem contínua não é exequível.

2.2 Recolha Periódica

Após se verificar que o processo de recolha de dados diminui significativamente a autonomia dos dispositivos, torna-se necessário solucionar ou minimizar este problema, delineando uma estratégia que permita manter o equilíbrio entre a eficácia da recolha e o consumo da bateria.

Como foi referido anteriormente, é importante maximizar o volume de dados recolhidos, no entanto garantir uma recolha eficiente de dados úteis não implica realizar uma amostragem constante e exaustiva dos interfaces. Em termos de deslocação humana, os dados são considerados úteis se permitirem perceber como as pessoas se deslocam num determinado espaço físico ao longo do tempo.

A forma mais simples de reduzir o consumo de energia é reduzindo a utilização dos interfaces, realizando uma amostragem periódica (Fig.4.7). Esta abordagem, implica a definição de um período de amostragem para cada interface, que permita manter a eficiência da recolha de dados.

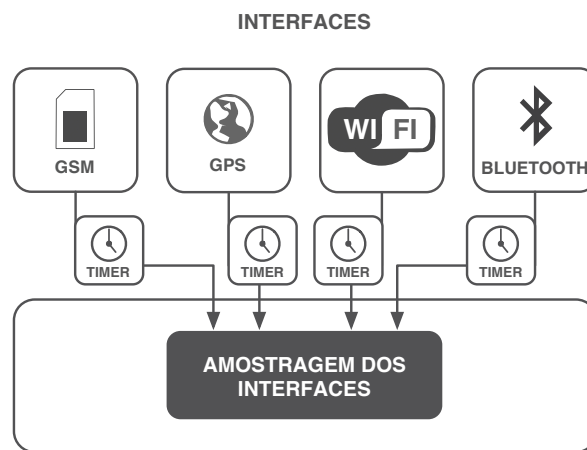


Figura 4.7: Amostragem periódica dos Interfaces

2.2.1 Período de Amostragem

A definição dos períodos de amostragem implica perceber quais os fatores que podem influenciar a recolha eficiente de dados. Os interfaces diferem em características, seja a nível de consumo de energia, do método de acesso, das informações disponíveis ou do alcance de funcionamento. Na figura 4.8, é possível observar que as diferenças de alcances variam de acordo com a tecnologia, sendo que os alcances máximos de cada uma dependem adicionalmente das

interferências e das características do meio envolvente.

Das tecnologias consideradas, o *Bluetooth* apresenta o menor alcance. Os dispositivos móveis incorporam módulos de *Bluetooth* Classe 2, que permitem um alcance de $\approx 10\text{m}$. O alcance máximo das redes *Wi-Fi* é $\approx 32\text{m}$ em ambientes fechados. No caso das estações base da rede celular, o alcance máximo é normalmente fixo e na ordem dos 35Km , sendo no entanto dependente do terreno ou do meio urbano, podendo ficar limitado a distâncias inferiores a 5Km , devido a prédios ou elevações do terreno nas imediações.

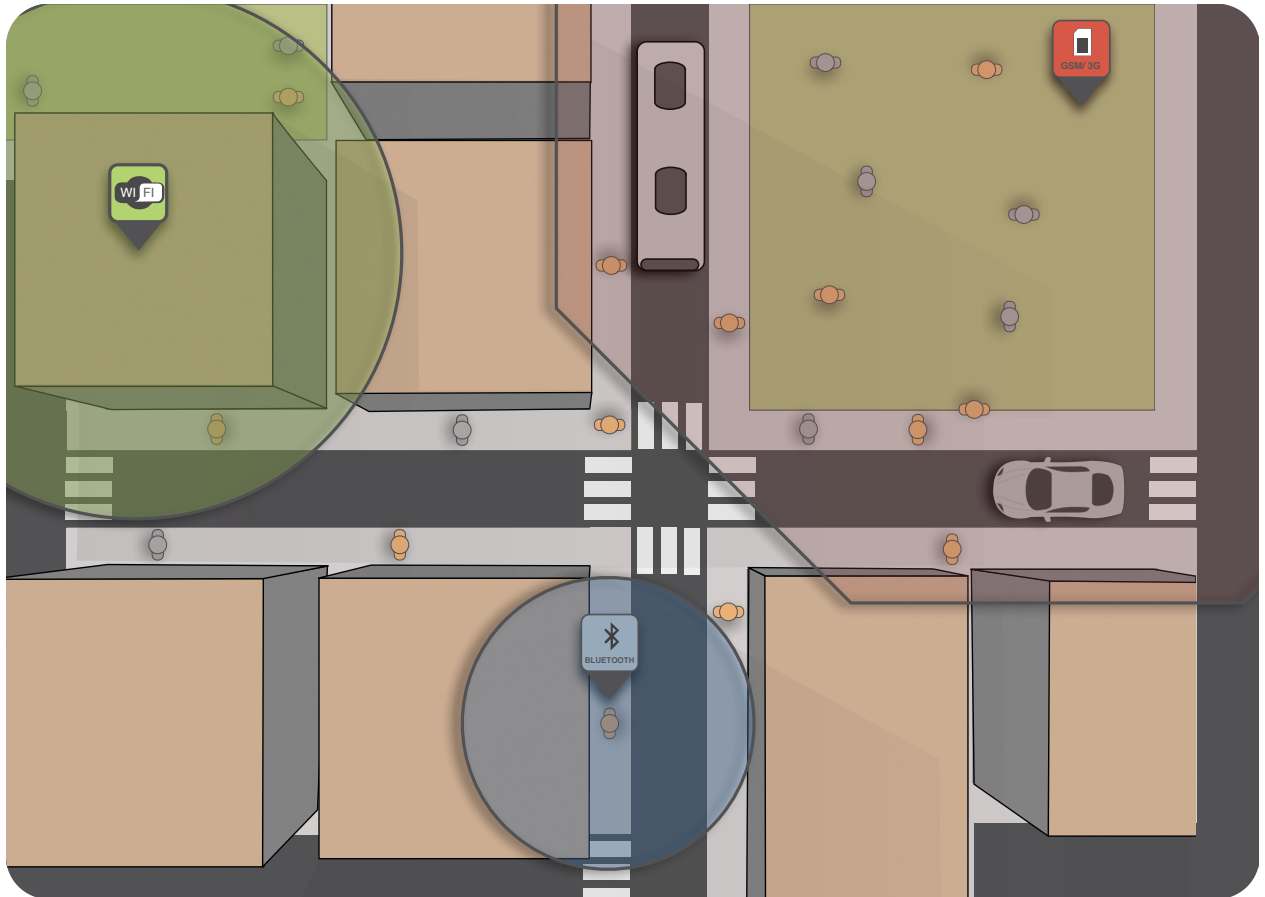


Figura 4.8: Diferenças de alcances entre tecnologias

As diferença dos alcances é, por isso, um dos critérios para a definição do período de amostragem. Considerando que o alcance do Bluetooth é inferior, revela-se necessário uma amostragem com periodicidade elevada, já que podem ocorrer alterações nos dispositivos detetados em curtos intervalos de tempo. No outro extremo temos de considerar o GSM porque devido ao seu elevado alcance, não ocorrem as alterações de forma tão frequentemente.

Outro critério a considerar, é a energia consumida na recolha dos dados de cada interface. Como mencionado anteriormente, a recolha de dados sobre a rede móvel celular não altera o consumo regular de energia do *smartphone*, assim este critério não influencia o seu período de amostragem. Já no caso do GPS, do Wi-Fi e Bluetooth, em que a sua utilização implica

consumos elevados, este critério deve ser considerado.

A tabela 4.2 resume os aspetos tidos em conta para a definição dos períodos de amostragem.

Interface	Consumo	Alcance	Tempo de Scan	Auto Scan
GSM	≈Nulo	Elevado(5-35Km)	-	-
Wi-Fi	Elevado	Médio(≈32m)	1.2 seg	6 em 6 seg
Bluetooth	Médio	Baixo(≈10m)	≈13 seg	-
GPS	Elevado	Não aplicável	11 seg	-

Tabela 4.2: Aspetos considerados para a definição dos períodos de amostragem

Segundo um estudo[KPN96] realizado, os utilizadores deslocam-se a pé durante uma parte considerável do dia, e a velocidade média de deslocação de uma pessoa a caminhar é de 1.46 m/s, sendo que muitas das deslocações dos utilizadores ocorrem de forma faseada. Considerando estes dados e as características dos interfaces, definiram-se períodos de amostragem, de forma a estimar os consumos de uma abordagem baseada em amostragem periódica. Períodos definidos:

- *Wi-Fi*: 10 segundos. Poderiam ser considerados os 6 segundos de *scan* automático. No entanto, devido ao alcance das redes *Wi-Fi*, os dados recolhidos de 6 em 6 segundos não seriam úteis, considerando que um utilizador dificilmente sai do alcance de uma rede Wi-Fi em 6 segundos.
- *Bluetooth*: 20 segundos. Considerando que o alcance do *Bluetooth* é inferior ao do *Wi-Fi*, à partida o seu período de amostragem deveria ser inferior. No entanto, o *Bluetooth* demora mais tempo a efetuar um *scan* e por isso um valor abaixo de 13 segundos não faria sentido.
- GSM: 20 segundos. Definiu-se este valor considerando que os consumos são praticamente nulos, podendo ser útil um período de amostragem curto para a deteção de estações base vizinhas.
- GPS: 60 segundos. Devido ao elevado consumo e ao tempo de *scan*.

Através da definição destes períodos de amostragem, estimou-se os consumos de bateria numa hora, sendo apresentados no gráfico 4.9. Comparando estes dados com as estimativas da amostragem contínua (Fig.4.6), denota-se uma redução significativa dos consumos (de 34.7%/Hora para 10.1%/ Hora), que se reflete no aumento da autonomia da bateria de ≈ 3 horas para ≈ 10 horas.

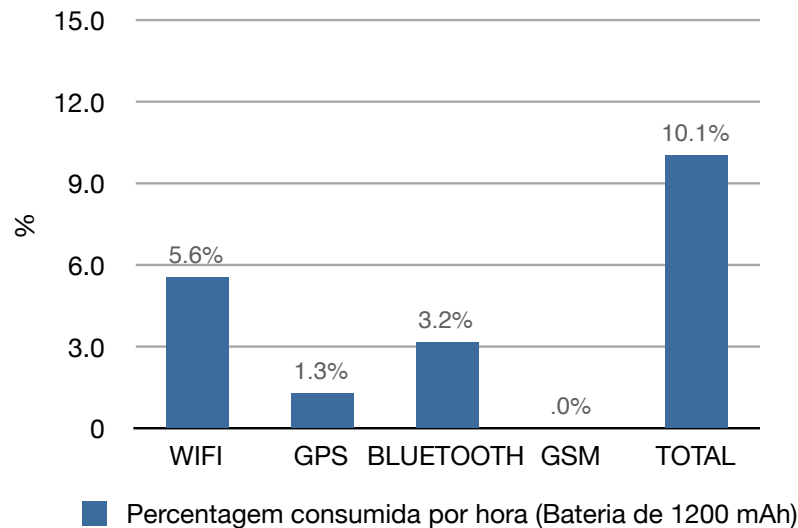


Figura 4.9: Estimativa da percentagem de bateria consumida por hora em recolha periódica

Apesar da redução dos consumos de energia ser significativa, os valores estimados ainda não se mostram aceitáveis.

Sendo que o período de amostragem influencia consideravelmente o consumo de energia, seria interessante uma análise mais aprofundada, identificando quais os períodos mais adequados para cada tecnologia, que permitem o melhor equilíbrio entre a eficácia da recolha de dados e o consumo de energia. No entanto, não faz parte dos objetivos desta dissertação discutir ou aprofundar mais esta questão. Em vez disso foram implementados mecanismos que possibilitam que as configurações da recolha de dados nos *smartphones* possam ser alteradas de forma remota. Estes mecanismos são explicadas no capítulo da implementação, na secção da recolha de dados. Deixa-se assim em aberto a possibilidade de períodos resultantes de estudos futuros possam ser aplicados na implementação atual.

2.3 Controlo da Recolha através do Acelerómetro

A amostragem periódica reduz a utilização dos interfaces de forma linear ao longo do tempo. Quando um dispositivo está estático, os dados recolhidos nesse período de tempo não são tão relevantes.

Segundo resultados de outros estudos, a percentagem média de movimento de uma pessoa durante um dia é inferior a 20% [CC11][CCR10]. Deste modo, se não forem recolhidos dados durante os períodos de inatividade ($\approx 80\%$ do dia), a redução dos consumos energéticos será significativa, aperfeiçoando a eficácia da recolha de dados, evitando a recolha de dados com menor utilidade.

Considerando a média diária de movimento de um utilizador (20%) e a percentagem de consumo da recolha periódica (10.1%/ Hora), conclui-se que em média existe uma redução de 8.1%/hora no consumo de bateria, refletindo-se no aumento da autonomia de 10 para 50 horas³.

Para a deteção do movimento utiliza-se o acelerómetro que está disponível em quase todos os *smartphones*, apresentando consumos reduzidos, como é possível observar no gráfico da figura 4.10.

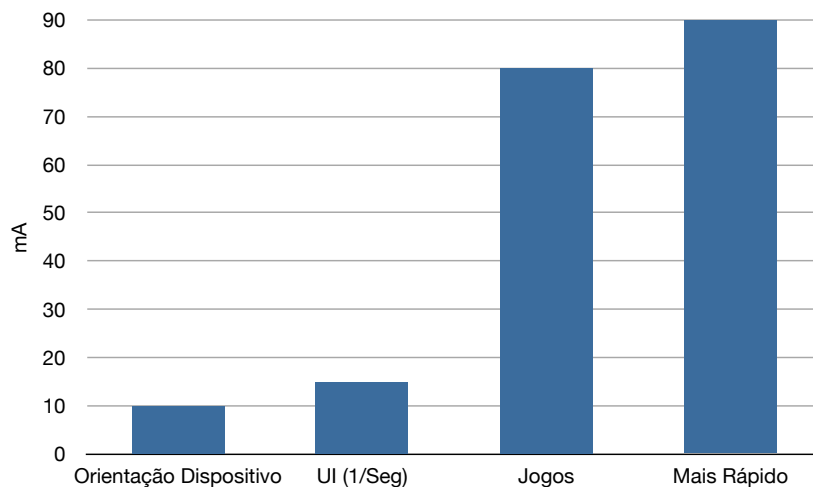


Figura 4.10: Carga elétrica utilizada pelo acelerómetro por unidade de tempo [Sha09]

O modo de funcionamento utilizado para a UI ou para a deteção da orientação do dispositivo é suficiente para detetar o movimento do dispositivo, implicando um consumo de apenas 15 mA. Este valor é consideravelmente inferior ao consumo de qualquer outro interface. Tendo em conta estas características, a utilização do acelerómetro é uma excelente abordagem para o controlo da recolha de dados. Como está representado na figura 4.11, efetuando a leitura e análise dos valores do acelerómetro, é possível identificar se o dispositivo está em movimento ou estático, e desta forma, iniciar ou pausar a recolha de dados.

³Não sendo considerados os consumos normais do funcionamento e de utilização do *smartphone*.

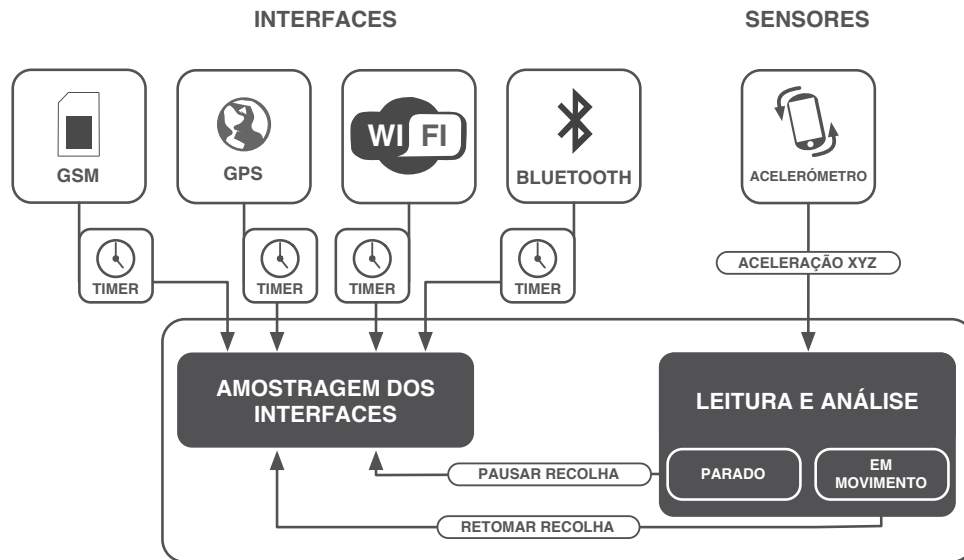


Figura 4.11: Recolha controlada pela detecção de movimento através dos acelerómetro

2.4 Conclusões

Através das estimativas realizadas, aliando o período de amostragem à utilização do acelerómetro para detecção do movimento do utilizador, alcançam-se consumos de energia aceitáveis (Fig. 4.12).

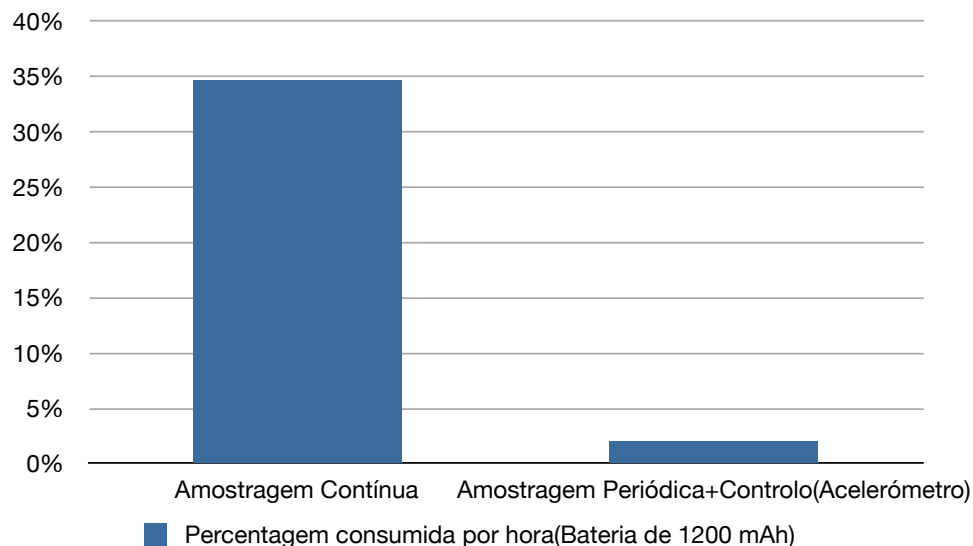


Figura 4.12: Comparação das estimativas da percentagem de bateria consumida por hora

É de realçar que estes valores são apenas uma estimativa dos consumos da utilização dos interfaces aplicando as estratégias referidas. É difícil, se não impossível, apresentar valores exatos, que, de forma geral, representem os consumos energéticos. Diferentes variáveis influenciam os resultados, como as características distintas dos *smartphones*, nomeadamente a capacidade da bateria e os consumos dos componentes e interfaces. Adicionalmente, os consumos totais

dependem do tempo de movimento de cada utilizador, assim como dos interfaces que se encontram ligados nesse período. As estimativas apontam para a disponibilidade de todos os interfaces ao longo do tempo (cenário de maior consumo). Na prática esta situação dificilmente se verifica.

Acresce aos valores estimados, o consumo do CPU no processo de recolha, que se demonstra difícil de contabilizar, dada a sua dependência do consumo do próprio *hardware* e das restantes aplicações que estão simultaneamente em execução.

A bateria considerada nas estimativas (1200 mAh), representa as baterias de menor capacidade nos *smartphones* disponíveis no mercado, direcionando as estimativas para consumos em dispositivos de baixo custo.

Apesar da dependência das variáveis citadas, os resultados demonstram que as soluções propostas se refletem positivamente na autonomia do dispositivo, podendo reduzir consideravelmente o problema do consumo de energia na recolha de dados.

Capítulo 5

Especificação do Sistema

1 Arquitetura do Sistema

A Arquitetura de Sistemas é uma disciplina genérica para lidar com objetos (existentes ou a serem criados) chamados sistemas, permitindo apoiar o raciocínio sobre as suas propriedades estruturais. Adicionalmente, surge como resposta às dificuldades conceptuais e práticas da descrição e design de sistemas complexos[Gol12].

Através da especificação da arquitetura é possível definir:

- A estrutura do sistema.
- Propriedades (dos vários elementos envolvidos).
- Relações (entre os vários elementos).
- Comportamentos e dinâmicas.
- Múltiplas visões do sistema (complementares e coerentes).

Os elementos principais do sistema são o servidor e o dispositivo móvel, que comunicam entre si através da Internet (Fig.5.1).

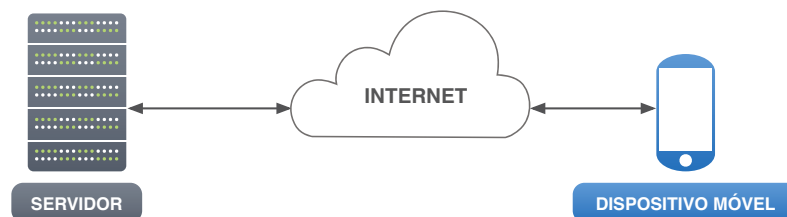


Figura 5.1: Servidor e dispositivo móvel

1.1 Arquitetura – Visão Geral

A aplicação recolhe informações através dos interfaces do dispositivo móvel. Desta forma, os elementos sobre os quais é recolhida informação fazem parte da arquitetura do sistema geral. Estes elementos são os seguintes:

- Satélites GPS, utilizados para determinar a posição do utilizador.
- Estações base GSM que estão no alcance do dispositivo.
- Pontos de acesso Wi-Fi detetados.
- Dispositivos Bluetooth nas proximidades.

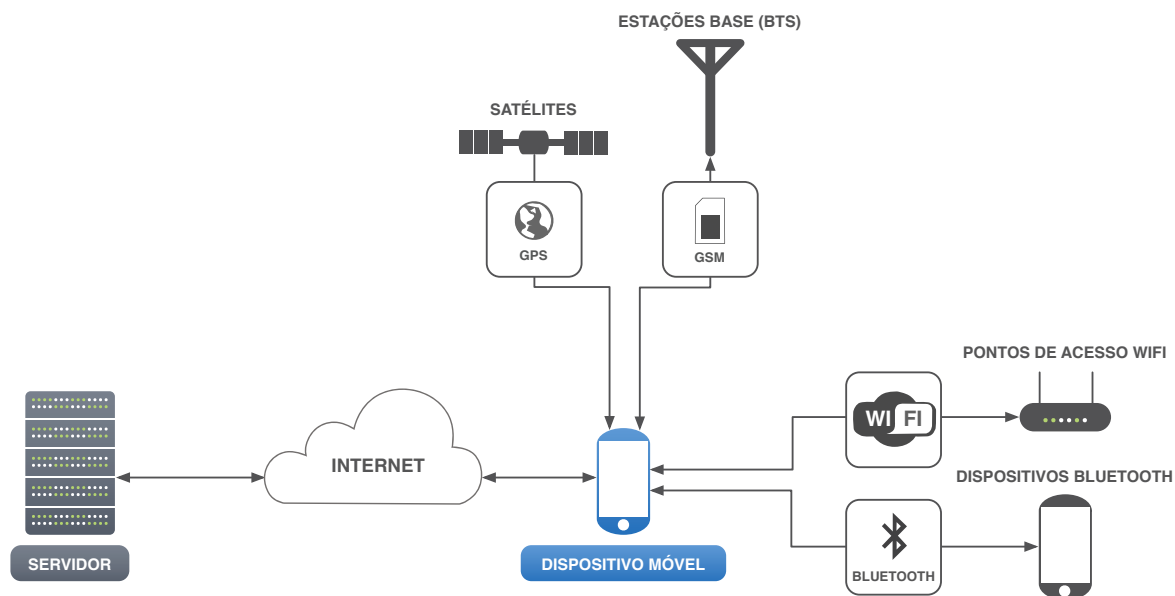


Figura 5.2: Arquitetura do sistema (Geral)

1.2 Análise de Requisitos

Em engenharia de sistemas e engenharia de software, a análise de requisitos engloba as tarefas para determinar as necessidades, as funcionalidades e as características de um sistema ou de um software.

Os requisitos podem ser distinguidos entre requisitos do sistema e requisitos do utilizador, sendo estes últimos requisitos de alto nível, relacionados com as funcionalidades disponíveis aos utilizadores. Os requisitos do sistema, normalmente descrevem pormenores técnicos e tecnológicos do sistema. Ambos subdividem-se em requisitos funcionais e não funcionais.

Os requisitos funcionais definem as funções de um sistema ou dos seus componentes, ou seja as funções que o sistema deverá desempenhar. Estes são suportados por requisitos não

funcionais, também conhecidos por requisitos de qualidade, que não alteram as funcionalidades essenciais do sistema, sendo normalmente referentes ao sistema geral e não a funcionalidades individuais. São considerados requisitos não funcionais, por exemplo o design de uma aplicação, assim como a sua usabilidade, fiabilidade, segurança e performance.

1.2.1 Requisitos – Recolha de Dados

Não sendo disponibilizada qualquer funcionalidade aos utilizadores, todos os requisitos da recolha de dados são requisitos do sistema.

■ Requisitos do Sistema

Requisitos funcionais:

- Fazer a recolha de dados utilizando os Interfaces Wi-Fi, Bluetooth, Rede Móvel Celular e GPS.
- Fazer o armazenamento temporário dos dados no *smartphone*.
- Fazer o envio dos dados para o servidor.
- Fazer a receção, processamento e armazenamento dos dados no servidor.
- Permitir configurações remotas.

Requisitos não funcionais:

- Garantir a segurança, privacidade e integridade dos dados.
- Garantir a segurança e privacidade dos utilizadores.
- Minimizar os consumos de energia.
- Otimizar a eficácia da recolha de dados.
- Otimizar o consumo de recursos no envio de dados, minimizando o *overhead*.

1.2.2 Requisitos – GeoAnúncios

Os requisitos do serviço e aplicação GeoAnúncios dividem-se entre requisitos do sistema e do utilizador.

■ Requisitos do Utilizador

Funcionais:

- Registo e autenticação.
- Publicação e gestão de anúncios georreferenciados.
- Consulta e procura de anúncios.
- Visualização dos anúncios através do mapa.
- Receção e configuração de notificações.
- Gestão de favoritos.
- Navegação para o local do anúncio.
- Resposta aos anúncios.
- Guardar anúncios por publicar.
- Envio automático de emails.

Não funcionais:

- Fluidez na visualização dos anúncios.
- Modo de funcionamento *offline*.
- Permitir o contacto com o anunciante mantendo a privacidade do seu email.
- Interface simples e visualmente atrativo.
- Facilidade na visualização e filtragem dos conteúdos.
- Sincronização dos conteúdos em vários dispositivos.
- Otimização do consumo de recursos no acesso ao serviço, minimizando o *overhead*.

■ Requisitos do Sistema

Funcionais:

- Permitir o acesso ao servidor através da Internet.
- Armazenar os conteúdos e contas de utilizador numa base de dados.
- Funções de gestão do serviço (Contas de utilizador e conteúdos).
- Notificação de novas atualizações da aplicação.
- Atribuição e gestão dos créditos.
- Gestão das associações dos dispositivos às contas de utilizador.
- Determinação de notificações de novos anúncios para cada utilizador.
- Gestão da disponibilidade dos anúncios segundo a data limite.
- Tornar os anúncios indisponíveis quando eliminados pelo anunciante.
- Manter *logs* de operações.

Não funcionais:

- Garantir a disponibilidade do serviço a vários utilizadores em simultâneo.
- Garantir a privacidade e segurança dos dados de registo dos utilizadores.
- Impedir o acesso aos conteúdos sem autenticação.

1.3 Módulos Aplicacionais

A programação modular é uma técnica de desenvolvimento de software, na qual as funções principais estão separadas em módulos, cada um dos quais contendo todos os elementos necessários para realizar a sua função independentemente[Haa12]. Os módulos representam uma separação de interesses ou objetivos, sendo a especificação dos módulos aplicacionais independente da linguagem ou da plataforma de implementação.

A aplicação móvel e o software do servidor a desenvolver, podem ser divididos em dois módulos principais, segundo dois objetivos:

- Prestação de um serviço ao utilizador.
- Recolha de dados de movimento.

A divisão (Fig. 5.3) confere ao sistema flexibilidade, dado que os módulos posteriormente podem ser utilizados de forma independente, ou ser integrados em outras aplicações. Desta modo, será futuramente exequível o desenvolvimento de aplicações móveis que fornecem outros serviços aos utilizadores, integrando o módulo de recolha de dados já desenvolvido.

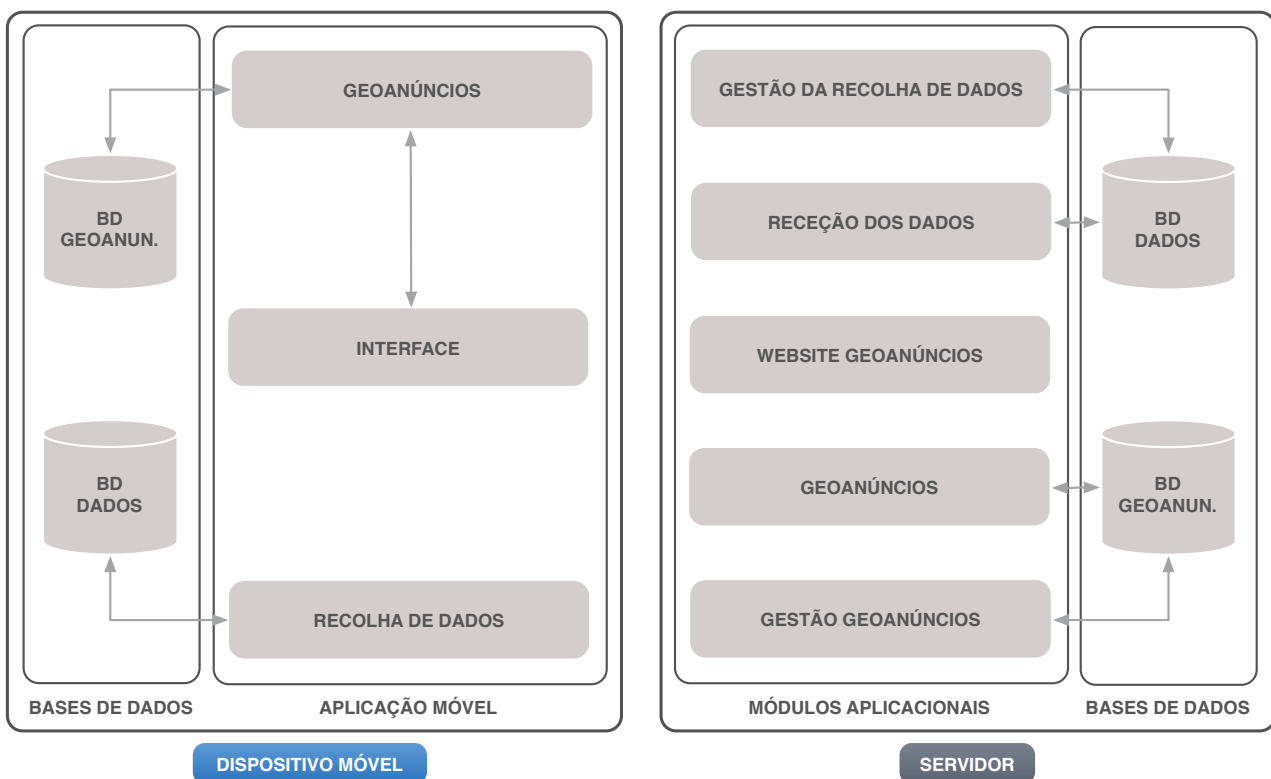


Figura 5.3: Módulos da aplicação móvel e servidor

1.3.1 Aplicação Móvel

■ Recolha de Dados

Este módulo inclui as funcionalidades necessárias à recolha de dados de movimento. Recebe periodicamente as configurações, efetuando a recolha de dados segundo as mesmas e armazena os dados recolhidos temporariamente, até que exista conectividade com servidor para o seu envio. Adicionalmente, controla o funcionamento da recolha de dados ativando e desativando a mesma através de um algoritmo de deteção de movimento e utiliza um algoritmo criptográfico para proteção dos dados enviados.

■ GeoAnúncios

De uma forma geral o módulo GeoAnúncios inclui todas as funções necessárias ao funcionamento da aplicação+serviço disponível ao utilizador. De modo a melhorar o desempenho e a experiência de utilização, armazena conteúdos localmente que permitem o acesso a determinadas funcionalidades do serviço, quando não existe ligação à Internet.

■ Interface

O interface concede ao utilizador a interação com o módulo GeoAnúncios. Apesar de muitas vezes desprezada a sua importância, principalmente em projetos experimentais, um interface com design visualmente apelativo tem elevado impacto na adesão dos utilizadores. Numa aplicação que pretende atrair utilizadores, é um fator quase tão determinante como as funcionalidades oferecidas.

Deste modo não é satisfatório utilizar apenas os elementos gráficos nativos da plataforma, é necessário criar elementos únicos que diferenciem a aplicação das restantes. Para a criação destes elementos podem e devem ser utilizadas ferramentas de design gráfico.

1.3.2 Servidor

■ Receção dos Dados

Este módulo é responsável por processar e armazenar os dados recolhidos e enviados pelas aplicações móveis e por guardar a configuração dos períodos de amostragem atual e seguintes de cada interface. O módulo envia para a aplicação ambas as configurações, assim como a data e hora a partir da qual entram em vigor. Adicionalmente, decifra os dados recebidos, utilizando o mesmo algoritmo criptográfico utilizado para cifrar na aplicação móvel.

■ Gestão da Recolha de Dados

O administrador, depois de devidamente autenticado no web site, pode alterar as configurações da recolha de dados e visualizar os dados recolhidos.

■ GeoAnúncios

Este módulo aplicacional é o núcleo do serviço GeoAnúncios, gerindo as contas de utilizador e os conteúdos disponíveis. Nas suas funções inclui-se a receção de termos para pesquisa e o retorno dos resultados, assim como conceder ao utilizador a gestão dos seus anúncios e da sua conta. O acesso às suas funções é controlado através da autenticação, sendo o registo a única de acesso livre. Este módulo é também responsável por informar as aplicações de novas atualizações e gerar *logs* de operações.

■ Gestão GeoAnúncios

É um módulo acessível através de um *website* que confere ao administrador, depois de autenticado, a gestão integral dos conteúdos do serviço.

■ Website GeoAnúncios

O web site oficial é importante para a divulgação. Neste, o utilizador pode descobrir mais informação acerca da aplicação e do projeto que lhe deu origem, através de conteúdos informativos. Disponibiliza a aplicação para *download*.

■ Bases de Dados

Os dados dos GeoAnúncios e os dados recolhidos são armazenados em bases de dados distintas. Esta separação permite que:

- O acesso às bases de dados seja independente: Pode ser necessário que outros serviços acedam aos dados armazenados, por exemplo no âmbito de outros estudos, não sendo necessário o acesso às informações dos GeoAnúncios.
- A privacidade dos utilizadores seja salvaguardada: Os dados dos utilizadores relativos ao serviço não estão associados de forma direta aos seus dados de movimento.
- Sistemas Modular: A alteração do conteúdo disponível para o utilizador é simples, não sendo necessário alterar a recolha de dados, e vice-versa.

1.4 Arquitetura do Sistema – Detalhe

A arquitetura especificada na figura 5.2 é uma abordagem geral ao sistema. Na figura 5.4 e figura 5.5 está representada a mesma arquitetura, sendo o seu funcionamento detalhado, figurando as relações entre os elementos e os seus comportamentos.

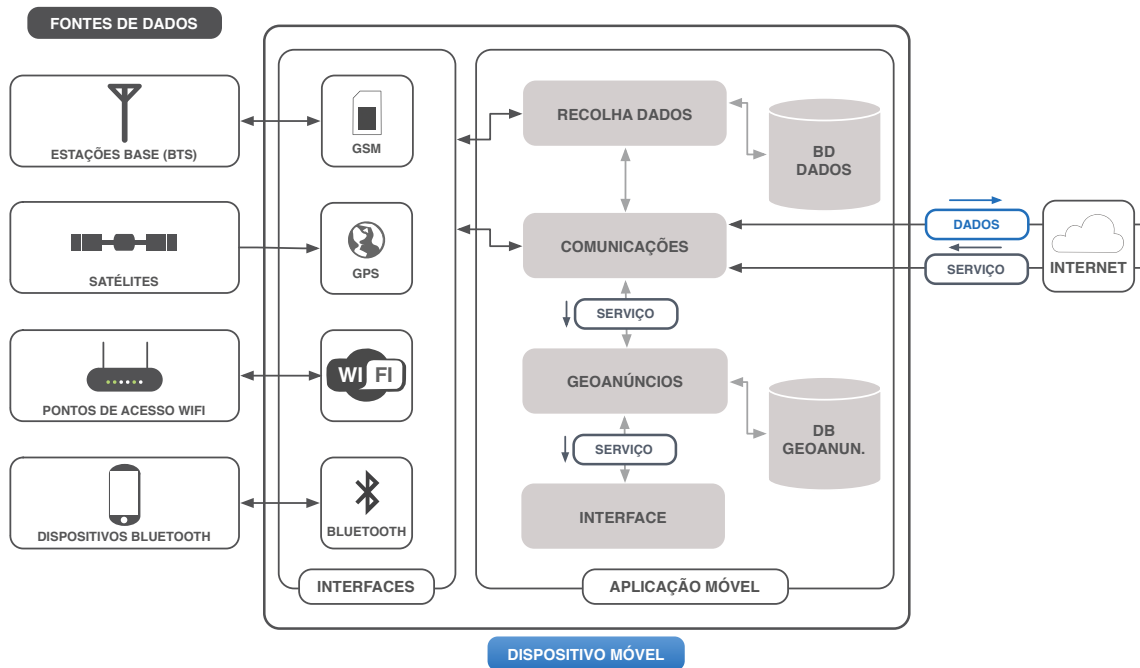


Figura 5.4: Arquitetura do sistema móvel (Detalhada)

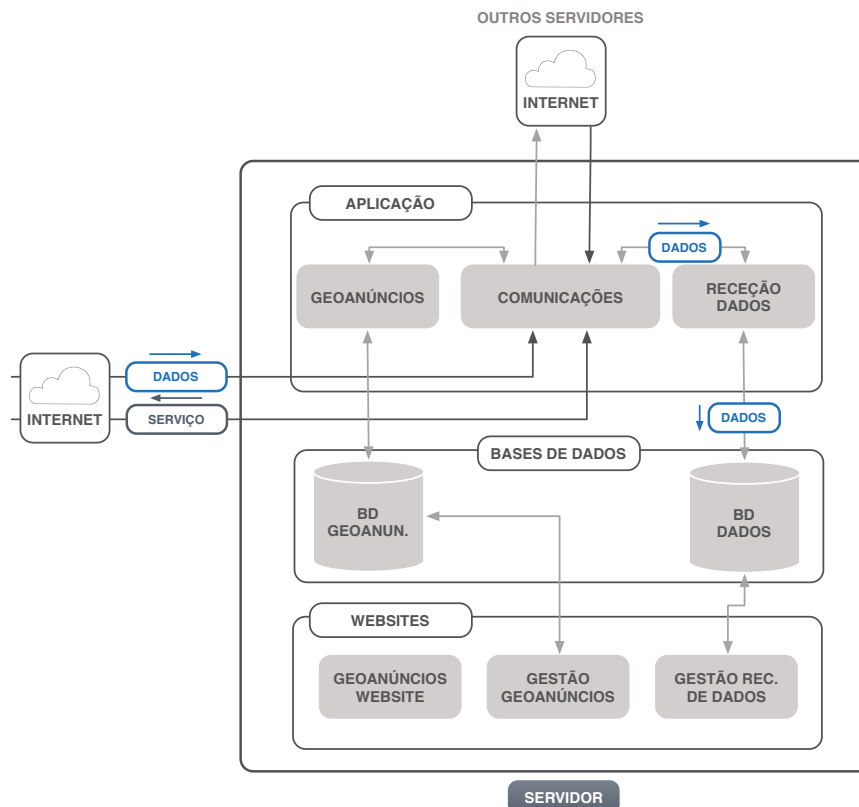


Figura 5.5: Arquitetura do servidor (Detalhada)

2 Mensagens do Sistema de Recolha de Dados

O envio dos dados recolhidos para o servidor e o envio das configurações de recolha do servidor para a aplicação, é baseado em duas mensagens. Para manter a aplicação atualizada no caso de serem necessárias alterações, existe uma mensagem de atualização.

■ Envio dos Dados Recolhidos para o Servidor

Quando os dados são enviados para o servidor, a mensagem contém as informações apresentadas na figura 5.6. Dependendo da fonte dos dados, ou seja do interface através do qual os dados foram recolhidos, as informações enviadas são diferentes (Tab. 5.1).

ESTRUTURA DADOS ENVIADA PARA O SERVIDOR					
SOURCE	DEVICE ID	DEVICE MAC	DATA	SCAN TIME	SEND TIME

Figura 5.6: Mensagem de envio dos dados recolhidos

- *Source*: Identifica o interface fonte da amostra (Wi-Fi, Bluetooth, GSM, GPS).
- *Device ID*: Identifica univocamente o dispositivo através do qual foi recolhida a amostra.
- *Device Mac*: É o endereço *Mac* do interface Wi-Fi ou Bluetooth, do dispositivo através do qual foi recolhida a amostra.
- *Scan Time* e *Send Time*: São respetivamente, a data e hora da recolha e de envio da amostra.

SOURCE	DATA				
GPS	LATITUDE	LONGITUDE	SPEED	ALTITUDE	ACCURACY
GSM	MCC	MNC	LAC	CELLID	RSSI
BLUETOOTH	MAC ADDRESS	NAME	CLASSE	RSSI	SERVICES
WIFI	BSSID	SSID	RSSI	CHANNEL	CAPABILITIES

Tabela 5.1: Dados enviados (Por Interface)

Em relação aos dados enviados por interface, em alguns casos, o próprio nome dos campos é suficiente para o seu entendimento, outros não são tão claros, nomeadamente:

- MCC: *Mobile Country Code*.
- MNC: *Mobile Network Code*.
- LAC: *Location Area Code*.
- RSSI: Nível de sinal recebido.
- *Capabilities*: Diz respeito às capacidades da rede, por exemplo, do nível de segurança.

■ Configurações da Recolha

Através da mensagem da figura 5.7, o módulo de recolha de dados da aplicação móvel recebe os valores dos temporizadores de cada interface, assim como a data e hora a partir da qual as mesmas devem ser utilizadas.



Figura 5.7: Configurações enviadas para a aplicação móvel

■ Informação de Versões

Na figura 5.8 está representada a mensagem recebida pela aplicação com as informações sobre a versão atual disponível no servidor. Esta mensagem permite alertar os utilizadores quando uma nova versão da aplicação é lançada.

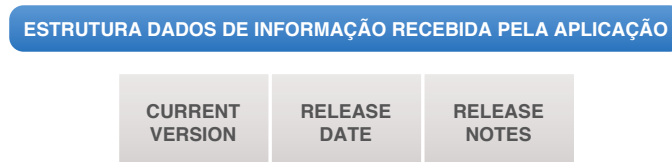


Figura 5.8: Informações sobre a versão atual da aplicação

O protocolo de comunicação do sistema encontra-se disponível de forma detalhada para consulta no apêndice C.

Capítulo 6

Implementação

1 Definição dos Parâmetros de Desenvolvimento

Após a especificação da arquitetura do sistema, serão agora apresentados os parâmetros de desenvolvimento, para a aplicação móvel e para o servidor.

Em primeira instância, foi realizada uma análise das plataformas, das tecnologias e dos recursos que poderiam dar resposta aos requisitos e desafios já identificados.

De seguida, são apresentadas e discutidas as vantagens e as desvantagens das alternativas consideradas para a implementação das principais componentes da aplicação móvel e do servidor, sendo fundamentadas as escolhas feitas. Conclui-se, apresentando os parâmetros de desenvolvimento resultantes.

1.1 Aplicação Móvel

Após a análise das diferentes plataformas móveis, tendo em conta as estatísticas de utilização, de crescimento e as suas características funcionais, a plataforma Android da Google e o iOS da Apple destacaram-se para o desenvolvimento da aplicação. Realçam-se das restantes pelas cotas de mercado e pelo crescimento no número de utilizadores ano após ano. Adicionalmente, disponibilizam documentação e ferramentas de desenvolvimento completas.

O ambiente de desenvolvimento (IDE) de aplicações para iOS, designado por XCode, disponibiliza um conjunto abrangente de ferramentas e a sua interface é atrativa, intuitiva e funcional. Realçam-se alguns pontos fortes em relação ao Eclipse, IDE oficial de desenvolvimento de aplicações para Android, nomeadamente a simplicidade, a disponibilização de ferramentas que conduzem à criação de interfaces visualmente apelativos de forma intuitiva e rápida. No Eclipse o processo para atingir o mesmo resultado demonstra-se mais demorado e trabalhoso, além disso são necessários conhecimentos de XML para controlar totalmente a criação do interface.

Contudo, o desenvolvimento da aplicação para iOS não foi possível, devido a condicionamentos tecnológicos que a plataforma impõe e ao modelo de negócio da Apple.

A disponibilização das aplicações para iOS é controlada pela Apple através de uma política rígida, que permite apenas a instalação de aplicações através da loja oficial App Store. O custo da licença para desenvolver e publicar aplicações na App Store da Apple (80 Euros/ ano), é uma desvantagem em relação à plataforma Android, em que o desenvolvimento é gratuito e a disponibilização no Google Play tem um custo de aproximadamente 20 Euros, valor pago uma única vez com duração vitalícia. Adicionalmente, dado que as aplicações para Android podem ser instaladas livremente, podem ser disponibilizadas através de um web site sem qualquer custo.

Além destas questões, o desenvolvimento para iOS foi inviabilizado por completo pelo facto da Apple impedir a utilização de APIs privadas para o acesso à recolha das informações das redes Wi-Fi, que são de elevada importância para este projeto. Em 2010, a Apple banuiu da App Store várias aplicações que utilizavam API privadas para obter as informações das redes Wi-Fi[And10], não permitindo no entanto a utilização da API nativa para que os *developers* tenham acesso a essa informação.

A Google não é tão rígida nos critérios de aprovação de aplicações para a sua loja online Google Play e fornece todas as APIs necessárias. Além disso as aplicações submetidas ficam rapidamente disponíveis aos utilizadores.

Acresce a estes factos o custo dos equipamentos, sendo possível adquirir um *smartphone* Android, equipado com *Wi-Fi*, *Bluetooth* e GPS, com razoáveis capacidades de processamento e memória, por valores que rondam ≈ 100 Euros. Em relação aos dispositivos iOS, não é possível adquirir um equipamento por menos de ≈ 490 euros[TMN12], não existindo uma gama de baixo custo. Apesar das características dos dispositivos Android mais baratos não poderem ser comparadas às do iPhone, para o desenvolvimento da maioria das aplicações, excluindo certos jogos que requerem outro tipo de *hardware*, os dispositivos mais acessíveis dão resposta às necessidades dos *developers*.

Após esta análise, a plataforma Android é a que melhor se adapta aos objetivos do projeto, sendo por isso a escolhida para o desenvolvimento da aplicação. Permite a utilização dos recursos tecnológicos necessários, o desenvolvimento e a disponibilização da aplicação de forma livre e gratuita, sendo ainda a plataforma com mais utilizadores e maior índice de crescimento no mercado. Os aspetos fundamentais da análise comparativa entre as duas plataformas encontram-se resumidos na tabela 6.1.

Aspectos	Android	iOS
Custo de desenvolvimento	Gratuito	Gratuito (Utilizadores de Mac OS X Lion e Mountain Lion)
Disponibilização	No Google Play (20€ uma vez) e por outros meios	Apenas na App Store (80.00€/ano)
Cota de mercado 1º Trimestre 2012 segundo a IDC	59%	23%
Permissão para utilização de todas as APIs necessárias	Sim	Não
Custo dos <i>smartphones</i>	Acessível. Desde ≈100€	Elevado. Desde ≈490€

Tabela 6.1: Resumo da análise das plataformas Android e iOS

1.1.1 Android – Arquitetura e Componentes

Após a eleição da plataforma Android para o desenvolvimento da aplicação móvel, é importante realizar uma análise mais pormenorizada da sua arquitetura e funcionamento (Fig. 6.1).

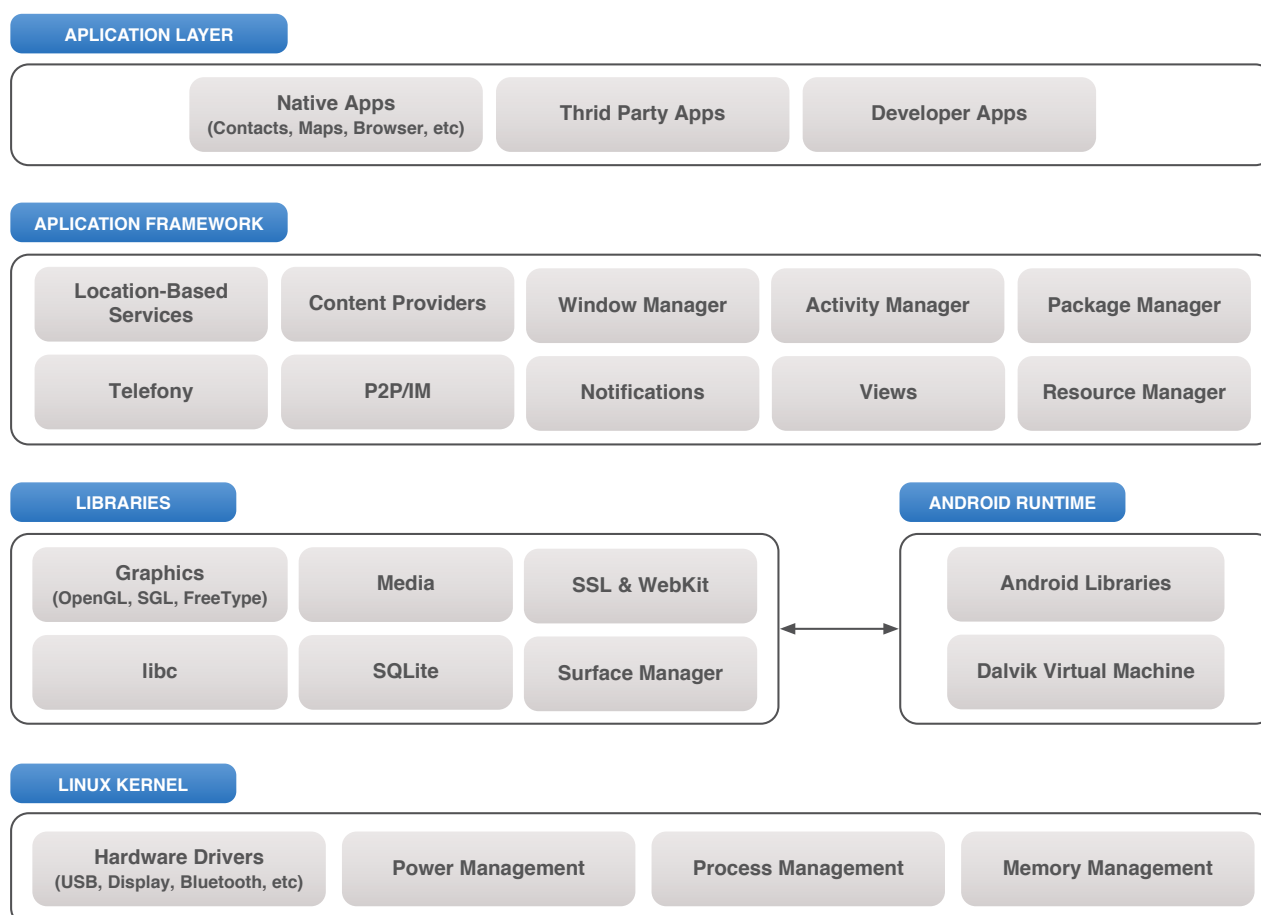


Figura 6.1: Arquitetura da plataforma Android[ARM12]

- **Linux Kernel:** Permite a abstração entre o hardware e o resto da *stack*. É responsável pelos drivers dos dispositivos (Bluetooth, Wi-Fi, etc), pela gestão de recursos, de energia, de segurança e de rede.
- **Libraries:** Bibliotecas nativas implementadas em C/C++, responsáveis pela gestão dos gráficos, reprodução de áudio e vídeo, SSL e WebKit para integrar segurança na navegação e suporte para bases de dados SQL Lite.
- **Android Runtime:** Inclui as bibliotecas *Core* que disponibilizam a maioria das funcionalidades da API Java e bibliotecas adicionais específicas do Android. Incluí ainda a máquina virtual Dalvik, que executa cada aplicação num processo separado, criando a sua própria instância da máquina virtual Dalvik.
- **Application Framework:** Fornece as classes necessárias para desenvolver uma aplicação Android e a abstração com o acesso ao *hardware*. A API Java da biblioteca principal do Android inclui os serviços de telefone, provedores de conteúdos (dados), gestão de atividades, gestão do ciclo de vida das aplicações, gestão de localização e de recursos UI, para a criação de interfaces.
- **Application Layer:** Todas as aplicações, nativas ou desenvolvidas/ descarregadas são criadas sobre a camada de aplicação, utilizando a mesma API.

■ Componentes de uma Aplicação Android

Uma aplicação Android é constitutiva por componentes. Os componentes, são os blocos essenciais de construção da aplicação, ligados através do manifesto do projeto. Cada componente desempenha um papel específico, ajudando a definir o comportamento geral da aplicação.

Cada tipo de componente tem uma finalidade distinta e um ciclo de vida diferente, que define como o componente é criado e destruído[And12b]. Existem quatro tipos diferentes de componentes:

- **Activities:** Uma *activity* ou atividade é implementada pela sub-classe *Activity*, representando um único ecrã ou menu da interface da aplicação com o utilizador. A maioria das aplicações têm várias *activities*. As *activities* são independentes umas das outras, apesar de quando unidas criarem ao utilizador a experiência de navegação na aplicação. A navegação para outro ecrã da aplicação consiste em iniciar uma nova *activity*. Para a navegação entre ecrãs é utilizada a classe *Intent*, que permite, para além de outras

funções, passar dados de uma *activity* para outra. Uma aplicação pode utilizar uma *activity* de outra aplicação, por exemplo uma aplicação pode utilizar a *activity* da galeria de fotografias de modo a importar uma foto.

- **Services:** Os serviços são utilizados para executar regularmente uma determinada tarefa, que deve continuar mesmo quando a *activity* ou a própria aplicação é terminada, como por exemplo fazer a verificação de novos emails. Um serviço não tem interface de utilizador e trabalha de forma independente da aplicação, no entanto é possível às aplicações comunicarem com o serviço.
- **Broadcast Receivers:** Permite que uma aplicação possa responder a um evento externo que ocorre no sistema, através da análise dos *Broadcast Intents*, que correspondem a um determinado filtro. Pode ser utilizado o *Notification Manager* para alertar o utilizador de algum evento sem interromper a *activity* atual. Uma aplicação não necessita de estar em execução para que um *Broadcast Receiver* seja acionado, apenas é necessário que o *Broadcast Receiver* esteja definido no manifesto da aplicação, para que este seja executado.
- **Content Providers:** Permitem armazenar e partilhar dados entre aplicações. Os dados podem ser armazenados em bases de dados SQLite. O Android permite o acesso a alguns *Content Providers* nativos, como por exemplo os contactos.

■ Ciclo de Vida das Aplicações Android

É importante entender o ciclo de vida das aplicações, de modo a definir a melhor forma de implementação do módulo GeoAnúncios e do módulo de recolha de dados. Dependendo dos componentes utilizados, a prioridade do processo executado pode ser:

- **Foreground** (Prioridade crítica): Quando o utilizador está a interagir com a aplicação. Neste estado o Android tenta manter a aplicação fluída atribuindo-lhe recursos.
- **Visible** (Prioridade alta): Neste estado, a aplicação perdeu o foco de interação, ficando em pausa. Por exemplo, quando uma *activity* que não utiliza o ecrã inteiro é chamada, a anterior continua visível. Neste estado, a *activity* mantém as informações, mas pode ser terminada pelo sistema por falta de recursos.
- **Service** (Prioridade alta): Os serviços não possuem interface com o utilizador, por este motivo recebem prioridade inferior em relação às *activities* em interação com o utilizador. No entanto, os serviços tem um elevado grau de importância, sendo apenas terminados em casos extremos de falta de recursos.

- **Background** (Prioridade baixa): São *activities* que estão completamente ocultas por outras, ou em *background*. Deste modo, podem ser terminadas sem impacto direto para o utilizador. O sistema quando necessita de recursos termina estes processos, utilizando uma lista que dá prioridade aos processos utilizados recentemente.
- **Empty** (Prioridade baixa): É um processo que não tem qualquer atividade. Estes processos podem ser mantidos como *cache* para que uma próxima iniciação seja mais rápida, sendo terminados sempre que necessário.

Após a análise do ciclo de vida das aplicações, foi possível perceber que a aplicação desenvolvida deveria ser baseada em dois tipos de processos. O módulo GeoAnúncios, baseado em *activities*, com interface de utilizador. A recolha de dados, executada como *Service*, obtendo assim uma prioridade alta e constante, sendo o seu funcionamento independente da aplicação GeoAnúncios.

Esta opção será novamente abordada, de forma pormenorizada, mais à frente neste capítulo, na implementação da recolha de dados.

1.2 Servidor

De modo a conceber uma solução para a implementação do servidor que dá suporte ao serviço GeoAnúncios e à recolha de dados, foi necessário dar resposta aos seguintes desafios:

- Garantir a disponibilidade através da Internet, de modo que a aplicação móvel possa aceder às funções implementadas.
- Gerir múltiplas comunicações em simultâneo. (Tanto para a receção dos dados recolhidos como para o serviço).
- Implementar mecanismos de segurança e de privacidade.
- Interoperabilidade do serviço, independente da plataforma do cliente. Tendo em conta que podem ser mais tarde desenvolvidas aplicações para outras plataformas móveis ou *Web*.
- Facilidade de acesso, APIs e protocolos de acesso simples e universais.
- Utilização de mensagens leves. Utilizando estruturas de dados simples e com pouco *overhead*.

1.2.1 Web Services – Introdução

Os *Web services* (por vezes referidos como serviços de aplicação), incluem normalmente uma combinação de programação e de dados, que são disponibilizados através de um *Web Server* para utilizadores *Web*, ou outro tipo de aplicações ligadas à *Web*. Enquanto as tradicionais aplicações *Web* apenas permitem a interação entre o utilizador e um *Web Site*, os *Web Services* permitem a comunicação entre aplicações, simplificando o acesso heterogéneo, independente da plataforma ou linguagem de programação utilizada no cliente.

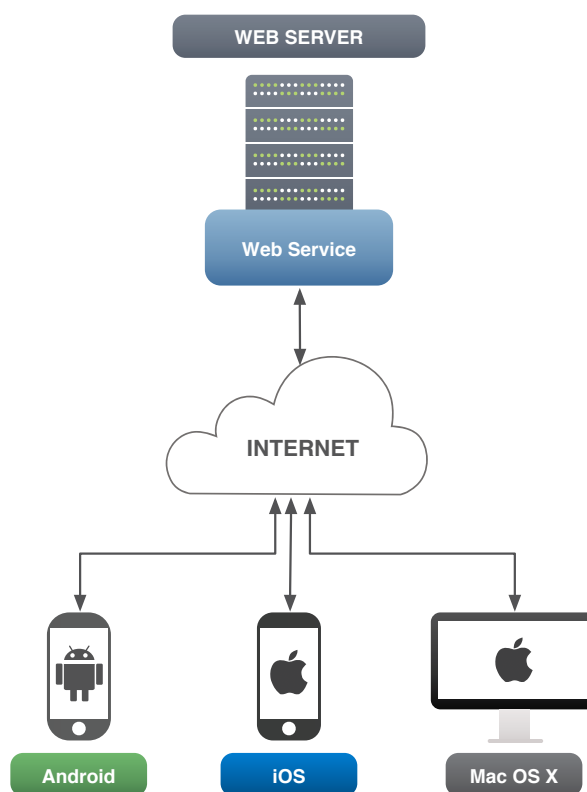


Figura 6.2: Web Service e clientes multi-plataforma

Os *Web Services* podem ser serviços complexos, como o armazenamento de dados online, ou serviços mais simples, como o fornecimento de informações sobre meteorologia ou sobre as cotações da bolsa. A criação e disponibilização deste tipo de serviços, tem sido uma das maiores tendências da *Web*[Rou07].

Os *Web Services* são sistemas de software desenhados para suportar a interação e interoperabilidade entre dispositivos através da Internet, utilizando geralmente o protocolo HTTP, que é o protocolo de aplicação da Internet mais utilizado. O protocolo HTTP é adotado universalmente, sendo um dos únicos protocolos aceites por praticamente todas as *Firewalls*.

Alguns *Web Services* podem comunicar com outros, essa troca de procedimentos e de dados, é geralmente realizada através de uma classe de software, conhecida como *middleware*. A *mid-*

dleware, pode ser descrita como uma camada de software interposta entre o sistema operativo e as aplicações, de modo a fornecer um nível de abstração e recursos úteis para o desenvolvimento de aplicações distribuídas. Os serviços que anteriormente apenas eram possíveis através do serviço padrão *Electronic Data Interchange* (EDI), estão cada vez mais a tornar-se *Web Services*.

Principais características de um *Web Service*:

- Os dados são trocados com base em linguagens padrão estruturadas, como por exemplo o XML.
- Integração de aplicações multi-plataforma pela Internet.
- Independência da linguagem de programação.
- Utilização de protocolos padrão, como o HTTP, que facilita a acessibilidade através das *Firewalls*.
- Possibilidade de grande variedade de clientes e funcionalidades, desde um simples pedido-resposta a uma transação comercial complexa.
- As aplicações podem ser facilmente transformadas em serviços acessíveis em qualquer lado, por qualquer dispositivo.
- Facilitam a integração de aplicações, utilizando uma infraestrutura leve, sem afetar a escalabilidade.

Muitos *Web Services* e APIs de acesso são desenvolvidos para interações servidor-servidor ou servidor-*browser*, não para aplicações móveis.

As plataformas móveis têm o seu próprio conjunto de desafios, devido à(s):

- Largura de banda.
- Memória e CPU disponíveis.
- Capacidade de armazenamento.
- Opções e questões de conectividade.
- Segurança e privacidade.
- Interação com o utilizador.

1.2.2 Web Services Interface – REST e JSON ou SOAP e XML

Para a interface com um *Web Service* através de HTTP pode ser adotada uma abordagem baseada em *Simple Object Access Protocol* (SOAP) ou em *Representational State Transfer* (REST). As vantagens e as desvantagens, assim como em que situações cada um deve ser adotado, têm vindo a ser discutidas.

■ Simple Object Access Protocol (SOAP)

O SOAP é um protocolo para troca de dados estruturados. Utiliza *eXtensible Markup Language* (XML) para o formato de mensagens e baseia-se geralmente em outros protocolos da camada de aplicação, principalmente no HTTP e SMTP, para negociação e transmissão de mensagens. De uma forma simples, o SOAP é um protocolo, baseado em XML, que permite às aplicações trocarem mensagens sobre HTTP.

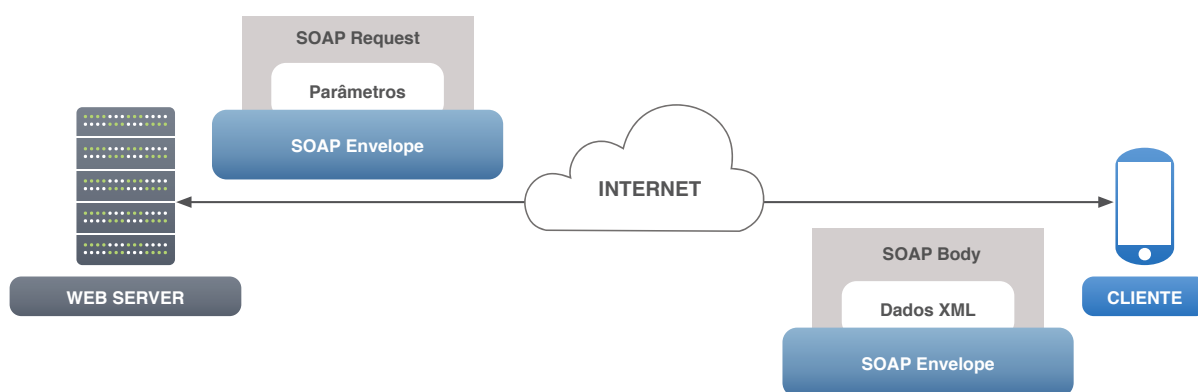


Figura 6.3: Mensagens SOAP invocam processos

■ Representational State Transfer (REST)

O REST tenta descrever arquiteturas que utilizam o protocolo HTTP ou similar, restringindo a interface a um conjunto de operações padrão conhecidas (como *GET*, *POST*, *PUT* e *DELETE*, no caso do HTTP). A interação é feita com recursos *stateless*, em vez de mensagens ou operações. As abordagens baseadas em REST utilizam o standard *Uniform Resource Identifier* (URI), que efetua um pedido ao *Web Service*, como por exemplo:

- `http(s)://www.empresa.com/programa/metodo?Parametro=x`

Esta abordagem é consideravelmente mais simples, podendo ser executada em qualquer cliente ou servidor que suporte HTTP(S). Um comando pode ser executado utilizando o método HTTP *GET* ou *POST*, por este motivo o desenvolvimento é menos complexo para os *developers* que

utilizam esta abordagem. A utilização da infraestrutura web existente e a sua simplicidade, são grandes vantagens[Roz10].

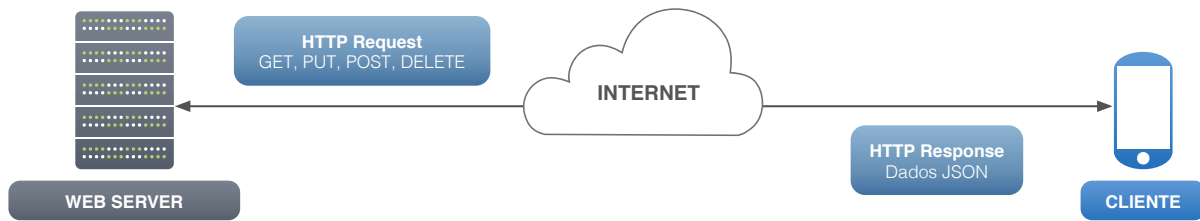


Figura 6.4: APIs REST obtêm recursos

■ *eXtensible Markup Language* (XML)

O XML é uma linguagem de marcação (*markup*)¹, que foi desenhada para transportar e armazenar dados. Define um conjunto de regras para o *encoding* de documentos interpretáveis por máquinas e humanos[w3s12a], identificando e dando sentido à informação.

O XML teve aceitação geral devido às linguagens de *markup* já estabelecidas, como o HTML. Graças à normalização pelo *World Wide Web Consortium* (W3C), foi rapidamente incorporado nas grandes ferramentas. Um das razões fundamentais para o sucesso do XML é a sua linguagem de definição de XML *Schemas* (XSD), que permitiu a sua utilização em inúmeros padrões como SOAP e WSDL[tec12].

Embora o XML tenha sido desenhado para documentos, é muito utilizado para a representação de estruturas de dados (por exemplo em *Web Services*). Muitas APIs foram desenvolvidos para utilizar e processar dados XML. Existem vários XML *Schemas* para auxiliar na definição de linguagens baseadas em XML, sendo desenvolvidas linguagens como RSS, Atom e SOAP com base no mesmo. As formatações em XML tornaram-se o padrão para muitas ferramentas de produtividade, como editores de texto[Cov12].

A utilização do XML na indústria tornou-se massiva. Sendo utilizado para armazenamento, transferência, apresentação ou configuração de dados e meta-dados[tec12]. No entanto, a sua utilização em APIs web tem vindo a decrescer significativamente nos últimos anos (Fig.6.5), muito devido à utilização de outros recursos que permitem as mesmas funcionalidade de forma mais simples e com menos *overhead*.

¹Uma linguagem de *markup* é uma linguagem que utiliza anotações, por exemplo *tags*, para que um dispositivo possa identificar e processar o texto.

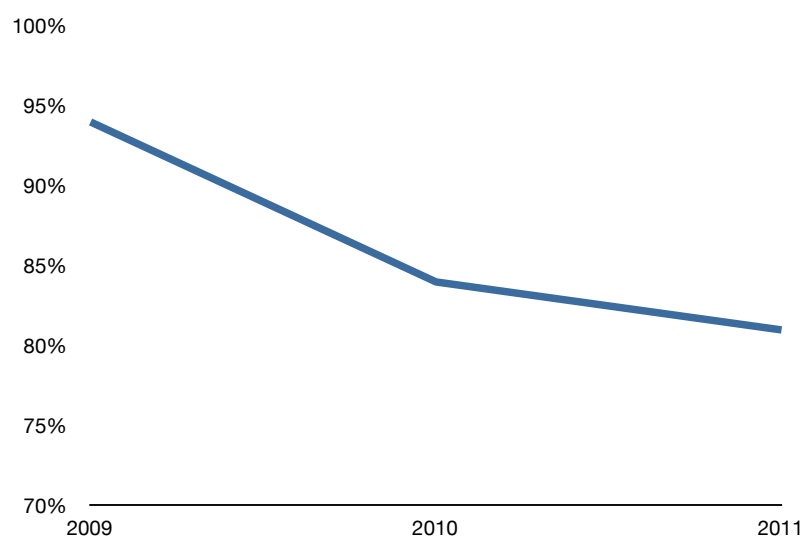


Figura 6.5: Percentagem de APIs com suporte XML.[Mus11]

Exemplo de dados em XML:

```
<peessoas>
  <pessoa>
    <nome>José</nome>
    <apelido>Gonçalves</apelido>
  </pessoa>
  <pessoa>
    <nome>Maria</nome>
    <apelido>Martins</apelido>
  </pessoa>
  <pessoa>
    <nome>Ana</nome>
    <apelido>Medeiros</apelido>
  </pessoa>
</peessoas>
```

■ *JavaScript Object Notation* (JSON)

O JSON é uma linguagem para o armazenamento e troca de informações de texto, semelhante ao XML. É aberta e estruturada, utilizando a sintaxe do JavaScript para descrever objetos de dados, no entanto não é dependente de uma linguagem de programação ou de uma plataforma. Existem bibliotecas e *parsers* de JSON para diversas linguagens de programação[w3s12b].

O JSON permite a fácil estruturação e *parsing* de dados e utiliza pouca largura de banda, o que aumenta a velocidade de transferência de dados.

O utilizador mais popular do JSON é a Google. Quase todas as operações dos serviços da Google (gmail, http, chat, Google Docs) utilizam exclusivamente JSON[adi12][Len09].

De todas as novas APIs que surgem, 55% suportam JSON² (Fig. 6.6)[Mus11].

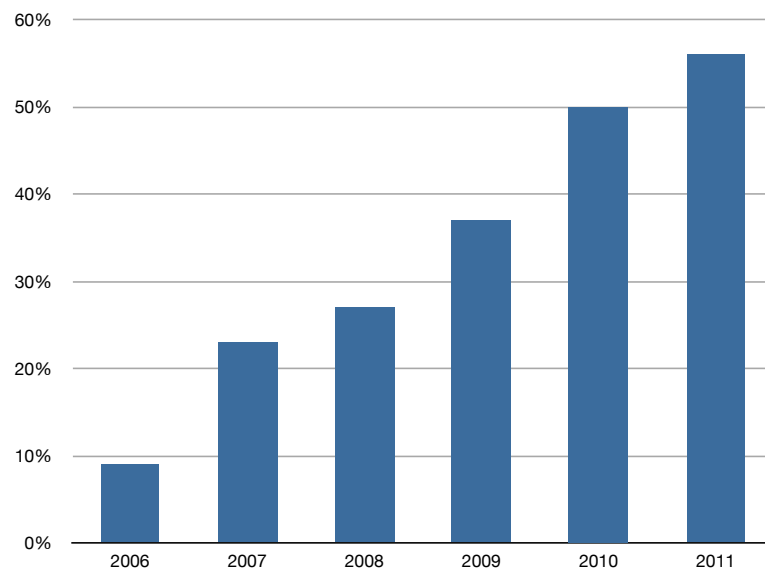


Figura 6.6: Percentagem de novas APIs com suporte JSON[Mus11]

Exemplo de dados JSON:

```
{
  "pessoas": [
    { "nome": "José" , "apelido": "Gonçalves" },
    { "nome": "Maria" , "apelido": "Martins" },
    { "nome": "Ana" , "apelido": "Medeiros" }
  ]
}
```

²Baseado no diretório de 3200 APIs listadas no *ProgrammableWeb*, Maio de 2011

■ SOAP vs REST

REST utiliza o HTTP padrão, sendo por isso significativamente mais simples em praticamente todas as situações. Seja na criação de clientes, no desenvolvimento de APIs, ou no entendimento da documentação. Não existem muitas situações onde o REST não dê resposta de forma mais simples/ melhor que o SOAP. As abordagens baseadas em REST permitem vários formatos de dados diferentes, o SOAP só permite a utilização de XML. Devido ao *parsing* do JSON ser significativamente mais rápido do que o do XML, a restrição do SOAP em apenas utilizar XML implica que este tenha mais *overhead* na transferência de dados.

O REST apresenta melhor desempenho e escalabilidade, permitindo ainda melhor suporte para os *browsers*, pois a maioria já incluem suporte nativo para JSON, e a abordagem REST utiliza os padrões *GET*, *PUT*, *POST* e *DELETE*. Adicionalmente, os pedidos REST podem ser armazenados em *cache*, enquanto o mesmo não é possível com SOAP.

Por estes motivos, a utilização de REST emergiu ao longo dos últimos anos como modelo predominante na criação de *Web Services*, devido à sua simplicidade em relação a modelos baseados em SOAP e WSDL. A Yahoo, Flickr utilizam REST para todos os seus serviços, a Amazon e Ebay disponibilizam ambos, no entanto internamente, a Amazon utiliza praticamente apenas REST. A Google utilizou somente SOAP para fornecer todos os seus serviços, mas em 2006, iniciou a alteração para REST e em 2009 anunciou o fim da sua API de pesquisa SOAP+XML, passando a disponibilizar apenas a nova API baseada em REST+JSON[Len09][Cov12].

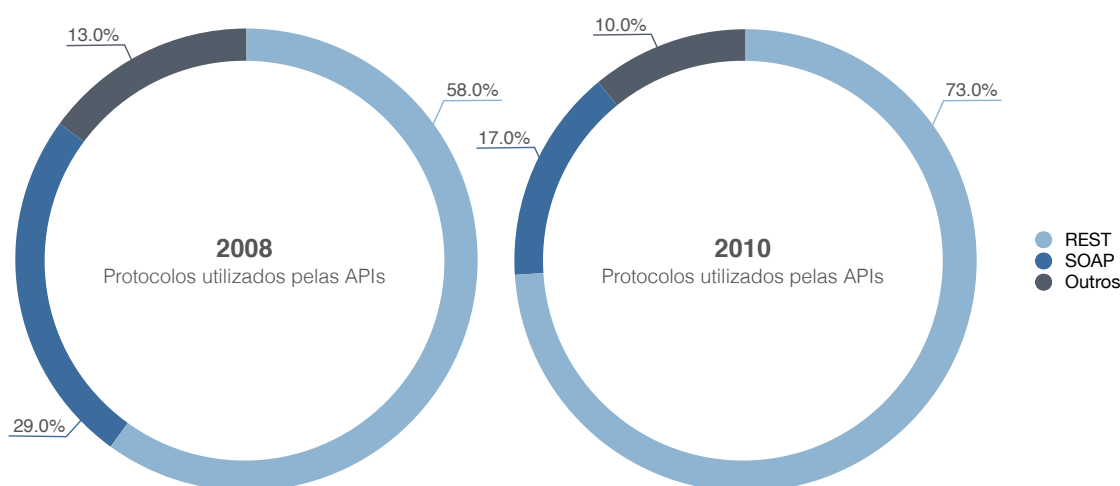


Figura 6.7: Distribuição dos protocolos utilizados pelas APIs[Mus11]

Apesar da abordagem baseada em REST ser a indicada na maioria das implementações, em determinados cenários o SOAP poderá introduzir vantagens relativamente ao REST. Ambos suportam ligações seguras através de SSL, contudo o SOAP suporta também *WS-Security*, o qual adiciona alguns recursos de segurança *enterprise*, suportando identidade através de

intermediários e não apenas ponto a ponto (SSL), disponibilizando ainda uma implementação padrão para garantia de integridade e privacidade de dados. O termo *Enterprise* não significa que é mais seguro, ou que o REST não é seguro, simplesmente suporta algumas ferramentas de segurança que apenas são realmente necessárias em alguns cenários empresariais, de que os serviços de Internet típicos não necessitam. No caso de serem necessárias transações que garantam *Atomicity, Consistency, Isolation, Durability* (ACID), é essencial a utilização de SOAP. O REST também suporta transações, porém não é tão abrangente, nem compatível com ACID. No entanto, transações ACID raramente são necessárias através da Internet, por isso a maioria das aplicações geralmente não precisa deste nível de confiança nas transações.

O REST não possui um sistema de mensagens padrão e os clientes têm que lidar com falhas de comunicação, efetuando uma nova tentativa. O SOAP, por sua vez, utiliza a lógica sucesso/repetição e permite confiança fim-a-fim, mesmo através de intermediários SOAP.

Em suma, o SOAP é claramente útil e importante. Por exemplo, no caso de uma aplicação móvel que faz o interface com um banco, todas as três características referidas são necessárias para as transações. Se for feita uma transferência monetária de uma conta para a outra, é necessário garantir que a mesma foi executada com sucesso. Voltar a efetuar a operação caso não exista uma confirmação de sucesso, como acontece no REST, pode ser desastroso se a primeira for concluída e apenas a resposta de confirmação falhar[Vin10].

Resumindo, o REST é indicado quando:

- Largura de banda e recursos limitados.
- As operações são *stateless*.
- É necessária *cache*. (A informação pode ser armazenada em *cache* devido à abordagem *stateless* do REST).

O SOAP é a solução para:

- Processamento e invocação assíncronos, se for necessário nível garantido de confiança e segurança nas transações.
- Contratos formais. Se ambos os lados (fornecedor e consumidor) tiverem de concordar com o formato de transferência, o SOAP fornece as especificações rígidas para este tipo de interação.
- Operações *statefull*. Se forem necessárias informações de estado e informações contextuais, permitindo suportar segurança, transações e coordenação.

■ XML vs JSON

As diferenças entre o XML e o JSON podem ser resumidas no seguinte: O XML é indicado para documentos e o JSON para troca de dados pela Internet.

Caso seja necessário lidar com documentos altamente estruturados e complexos, o XML é sem dúvida a ferramenta indicada. Se o objetivo é a troca de mensagem que ocupem pouca largura de banda e que o seu *parsing* seja simples e rápido, o JSON é o ideal[tec12], já que o JSON não implica tanto *overhead* como o XML, como foi possível observar pelos exemplos das suas estruturas apresentados anteriormente.

Resumindo, o XML é:

- Centrado em documentos.
- Bem conhecido por uma grande variedade de ferramentas.
- Pesado.

O JSON é:

- Centrado em estruturas de dados simples.
- Não é tão conhecido.
- Leve. Permite poupar largura de banda e não exige tantos recursos computacionais, já que o seu *parsing* é rápido e simples.

■ Conclusões

As razões para a abordagem REST+JSON ser adotada em detrimento da abordagem SOAP+XML pode ser resumida pelo seguinte:

- O JSON consome menos largura de banda e exige menos processamento. Nos dispositivos móveis estas características fazem toda a diferença, devido às suas limitações de hardware e energéticas. Quanto menor for o processamento e o tempo de transferência de dados, menor é o consumo de bateria. A redução do consumo de largura de banda é importante, principalmente quando utilizados os dados móveis através da rede móvel celular.
- Os interfaces REST são consideravelmente mais simples de projetar e implementar, pois o REST utiliza o padrão do HTTP(*GET*, *POST*, *DELETE*), deste modo já estão definidas todas as semânticas e controlos de semântica. O *developer* apenas necessita de se concentrar na modelação das estruturas de dados em JSON e na modelação das hierarquias URL. O melhor desempenho e escalabilidade do REST é igualmente uma vantagem, permitindo um elevado número de pedidos simultâneos, ou seja o suporte de um elevado número de utilizadores.

Sendo que a aplicação móvel não necessita de executar operações ACID, uma abordagem *stateless* baseada em REST é a solução ideal.

A interface com o *Web Service* foi assim baseada numa abordagem REST+JSON utilizando o protocolo HTTP para a transferência dos dados. Esta solução é a que melhor se adapta às características e requisitos do sistema a desenvolver, permitindo o interface simples entre a aplicação móvel e o *Web Service*. Além disso, através da adoção desta abordagem foi dado o primeiro passo na resolução de um desafio importante deste projeto, que é a redução do consumo de energia no dispositivo móvel.

1.2.3 Web Service Engines

Se excluídos o C e o C++, as linguagens de programação mais populares são o Java e o PHP, constituindo a maioria dos serviços *web*[Dai11].

A linguagem Java foi desenvolvida pela Sun Microsystems em 1995, embora originalmente destinada para o *software* cliente e para *applets in-browser*, é hoje uma infraestrutura essencial para muitas aplicações *web*.

O PHP nasceu no mesmo ano e foi criado especificamente para a web, sendo uma linguagem de *script* para o lado do servidor, capaz de ser integrada em páginas HTML. Este evoluiu através de cinco versões principais e tornou-se uma das linguagens de programação mais populares da web, graças também a serviços de partilha e armazenamento de dados, que são particularmente simples de configurar para aplicações PHP, permitindo reduzir os custos.

O PHP é conhecido como uma linguagem de programação da era da Web 2.0, que permite o desenvolvimento flexível de software do lado do servidor. Um elevado número de software *open-source* e web sites de empresas utilizam PHP, já que permite elevada produtividade.

O Facebook, a Wikipedia, o Wordpress e a Yahoo, são apenas alguns exemplos dos gigantes da web que utilizam PHP[W3T12].

De acordo com o site dailychanges.com³[Dai12], a 30 de Agosto de 2012 existiam perto de 141 milhões e 700 mil domínios registados. Segundo o W3Techs.com[W3T12], o PHP é utilizado por 78% de todos os *web sites* que utilizam linguagens de programação do lado do servidor (Fig.6.8⁴).

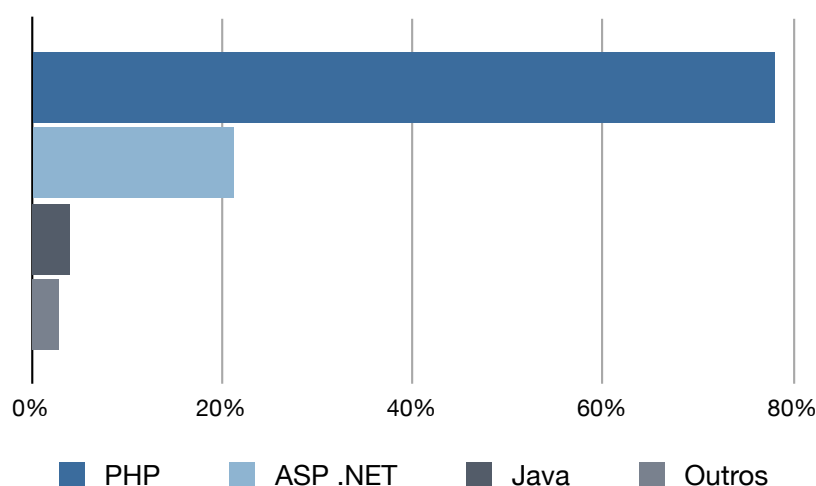


Figura 6.8: Percentagens de websites que utilizam as diferentes linguagens de programação

³DailyChanges.com monitoriza as mudanças de DNS e apresenta informação significativa sobre essas mudanças.

⁴Um web site pode utilizar mais do que uma linguagem de programação do lado do servidor.

Existe por vezes um estereótipo em relação ao PHP, como tendo um tempo de execução moderado, por ser uma linguagem interpretada. No entanto, o PHP pode armazenar em *cache* o *opcode* intermediário na *Alternative PHP Cache* (APC). Além disso o conjunto padrão de bibliotecas de extensão são implementadas principalmente em C. Estas características distinguem o PHP das outras linguagens de *script* dinâmicas como o Ruby e o Python.

A linguagem PHP é considerada como indicada para a implementação de serviços REST, sendo bastante simples implementar este tipo de serviços em PHP. A utilização dos *scripts* desenvolvidos é tão simples como localizar o *script* PHP no diretório web apropriado.

O Java é normalmente implementado como *servlets* num *engine servlet*, como por exemplo o Apache Tomcat.

Teste realizados por Suzumura et al[STT⁺08], em que foram comparadas, entre outras, as duas configurações representadas na tabela 6.2⁵, evidenciam que a configuração Apache+PHP5 apresenta uma performance superior em aproximadamente 70% à da configuração Tomcat+Axis2 Java(Fig.6.9).

Configuração	Servidor HTTP	Conector	Web Services
Axis2 Java	Tomcat 5.5.26 (Java)	Nenhum (Thread)	Java
PHP+Apache	Apache 2.0.63	mod_php	PHP

Tabela 6.2: Configurações utilizadas para o teste[STT⁺08]

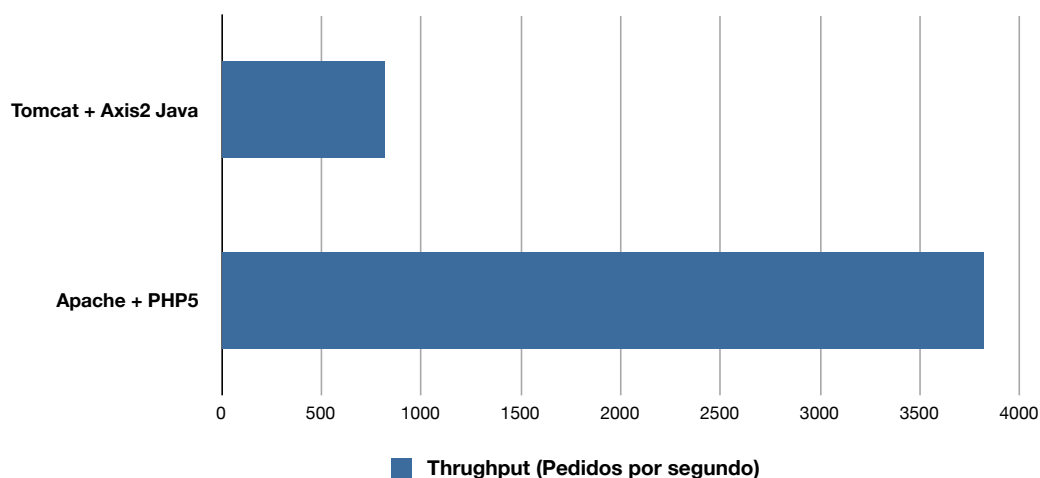


Figura 6.9: Resultados do teste Tomcat+Axis2 Java e Apache+PHP5[STT⁺08]

⁵Web server dinâmico sem SOAP engine.

O PHP utiliza na execução somente o que necessita, o que representa uma vantagem em termos escalabilidade, no entanto devido a este facto é incapaz de executar tarefas periódicas. As aplicações Java podem iniciar várias *threads*, mas a sua gestão é muito mais complexa, desde a fase de compilação à de implementação.

Em relação à infraestrutura *web*, o PHP é simples de implementar na sua forma básica (*scripts .php*), no entanto o programador tem por vezes de utilizar *frameworks* que constroem outros recursos sobre o interpretador PHP. Ironicamente, estes *frameworks* são semelhantes aos do Java (*servlets*), classes com um construtor padrão que estendem uma classe base comum. O Java tem menos funcionalidades web nativas integradas na linguagem, visto que não é estritamente orientada para a web, contudo possui as mesmas em *frameworks*, os *servlet containers*.

■ Conclusões

Após a conclusão desta análise, optou-se por utilizar o PHP como linguagem de desenvolvimento dos módulos do lado do servidor, sendo adotada como *Web Service Engine*.

O PHP possui as capacidades necessárias para lidar com as bases de dados. Por ser uma linguagem criada especificamente para a web, praticamente todas as funções necessárias para a criação de um *Web Service* estão disponíveis de forma nativa, podendo ser também utilizados *frameworks* complementares. É uma linguagem que oferece elevado *throughput* e utiliza na execução apenas o que necessita.

Adicionalmente, o PHP pode ser integrado de forma simples com *HTML*, permitindo desta forma que os *Web Services* desenvolvidos em PHP possam ser utilizados também através de um web site, criando opções interessantes. É possível, por exemplo, fazer a gestão do serviço Geo-Anúncios ou do módulo de recolha de dados sem ser necessário criar outros módulos específicos para gestão via web site.

Outro cenário interessante é a disponibilização do serviço GeoAnúncios também através de um web site, em que o utilizador da aplicação móvel poderá gerir ou visualizar os conteúdos do serviço, mais uma vez sem necessidade de alterações nos módulos que prestam o mesmo.

1.3 Características e Parâmetros do Sistema – Resumo

■ Aplicação Móvel

- Plataforma: Android
- Linguagem de desenvolvimento: Java
- Android SDK: Versão 2.2 (API *level* 8)⁶
- Ambiente de desenvolvimento (IDE): Eclipse (Mac OS X)

■ Servidor

- HTTP Web *Server*: Apache 2.2 (Windows)
- Web *Service Engine*: PHP 5
- Web *Service API/ Interface*: REST+JSON
- Base de dados: MySQL

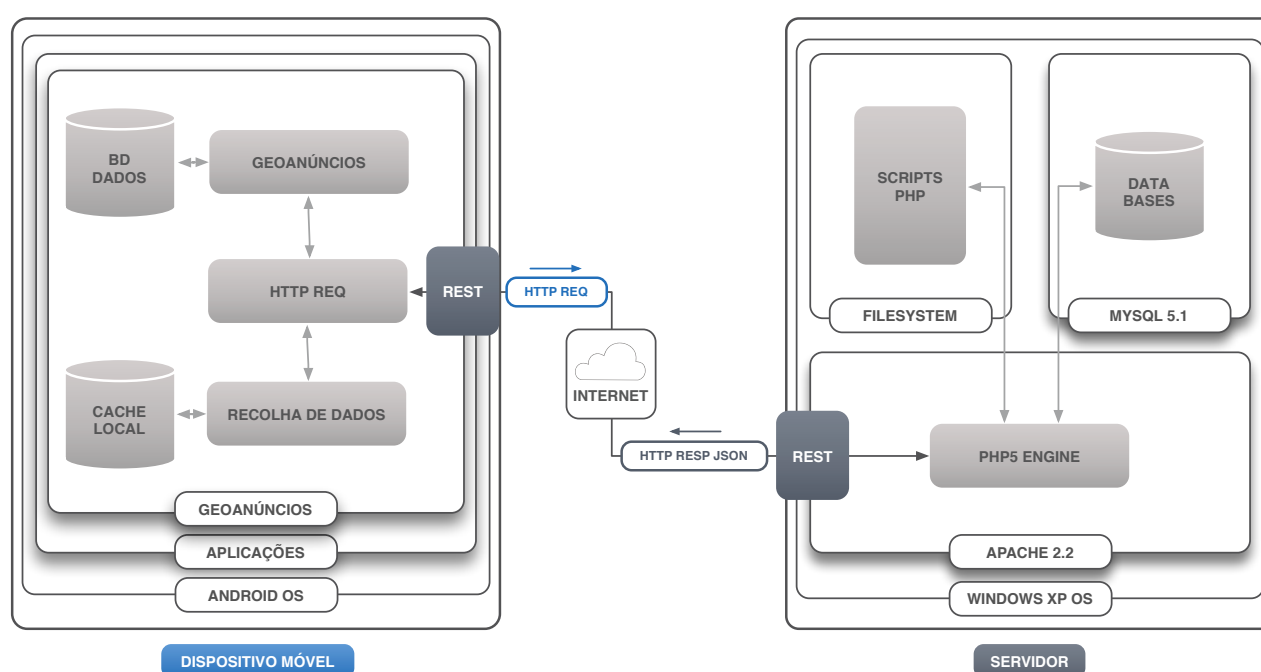


Figura 6.10: Representação das tecnologias do sistema

⁶Optou-se pelo desenvolvimento com o SDK 2.2, de forma a suportar todos os dispositivos com versão do Android 2.2 ou superior.

2 Recolha de Dados – Módulo *DataManager*

O processo de implementação iniciou-se com o desenvolvimento do módulo que permite a recolha de dados no *smartphone*, denominado de *DataManager*. A sua implementação pode ser dividida em oito fases:

- Fase 1: Análise das APIs do Android que permitem o acesso aos interfaces do *smartphone* (GPS, Rede Móvel Celular, *Bluetooth*, *Wi-Fi*), identificando as informações que é possível recolher.
- Fase 2: Implementação e teste de módulos de recolha de dados individuais para cada interface.
- Fase 3: Implementação de períodos de amostragem.
- Fase 4: Armazenamento local dos dados recolhidos.
- Fase 5: Envio oportunista dos dados.
- Fase 6: Receção das configurações do servidor e controlo do processo de recolha através das mesmas.
- Fase 7: Transformação do módulo *DataManager* em serviço Android.
- Fase 8: Controlo da recolha de dados através de um algoritmo que utiliza os valores do acelerómetro para detetar o movimento.

2.1 Recursos Utilizados e Dados Recolhidos

A recolha de dados é efetuada com recursos às APIs[And12a] nativas do Android apresentadas na tabela 6.3, que permitem o acesso/ controlo dos interfaces do *smartphone*. Adicionou-se a recolha do nível da bateria, que pode revelar-se interessante posteriormente para análises mais aprofundadas dos consumos energéticos nos *smartphones*.

Interface	API	Descrição
WI-FI	android.net.wifi	Permite o acesso às informações relativas ao interface Wi-Fi. Incluindo a velocidade de ligação, o endereço IP, Mac Address, assim como informações sobre outras redes disponíveis.
Bluetooth	android.bluetooth	Fornece classes que permitem gerir o Bluetooth, tais como a procura de dispositivos, as conexões com dispositivos, e transferência de dados entre dispositivos.
GPS	android.location	Permite obter as coordenadas GPS atuais do utilizador, entre outras informações.
Rede Móvel	android.telephony	Permite obter informações sobre o tipo de rede, estado da ligação e das estações base.
Bateria	android.os.BatteryManager	Permite recolher a informação do estado da bateria, sempre que a percentagem de bateria muda.

Tabela 6.3: APIs Android utilizadas para acesso às informações das Interfaces

Através da utilização das APIs anteriores foi possível recolher os dados seguintes.

2.1.1 Pontos de acesso *Wi-Fi*

- BSSID: O Endereço do ponto de acesso.
- SSID: O nome da rede.
- *Capabilities*: Descreve a autenticação, a gestão de chaves e os esquemas de criptografia suportados pelo ponto de acesso.
- *Channel*: A frequência, em MHz, do canal pelo qual o cliente está a comunicar com o ponto de acesso.
- *Received Signal Strength Indication* (RSSI): O nível de sinal detetado, em dBm.

2.1.2 Dispositivos *Bluetooth*

- *Mac Address*: Endereço do interface Bluetooth do dispositivo detetado.
- *Name*: Nome definido pelo utilizador ou nome por defeito do dispositivo.
- *Classe*: Classe do dispositivo (Telefone, computador, etc).
- *Services*: Serviços disponíveis no dispositivo (*Audio*, *Networking*, etc).
- RSSI: Nível de sinal detetado em dBm.

2.1.3 Estações Base

- *Mobile Country Code* (MCC): Identifica univocamente um país.
- *Mobile Network Code* (MNC): Identifica univocamente uma operadora.
- *Location Area Code* (LAC): Identifica uma área.
- CELLID: É a identificação da estação base.
- RSSI: É o nível com que o sinal da estação base é recebido.

2.1.4 Posição GPS

- Latitude: Coordenada geográfica que especifica a posição norte-sul de um ponto sobre a superfície da Terra
- Longitude: Coordenada geográfica que especifica a posição leste-oeste de um ponto sobre a superfície da Terra
- *Speed*: Velocidade de deslocação do dispositivo.
- Altitude: Altitude do dispositivo.
- *Accuracy*: Exatidão, em metros, da posição recolhida.

2.1.5 Nível de Bateria

- Nível atual da bateria, de 0 a 100%.

A cada amostra recolhida de cada interface, é adicionada a data e hora da recolha.

2.2 Bases de Dados

Nas bases de dados para armazenamento das amostras recolhidas (Fig. 6.11 e 6.12), por opção não se estabeleceram relações entre as tabelas. A associação dos dados recolhidos em cada *scan* é conseguida através do campo "DATA DE SCAN". Os campos com "DATA", referem-se a um campo com o formato ano-mês-dia hora:minutos:segundos.

■ Aplicação Móvel



Figura 6.11: Diagrama Entidade Relação (DER) da BD dados recolhidos (Aplicação Móvel)

■ Servidor

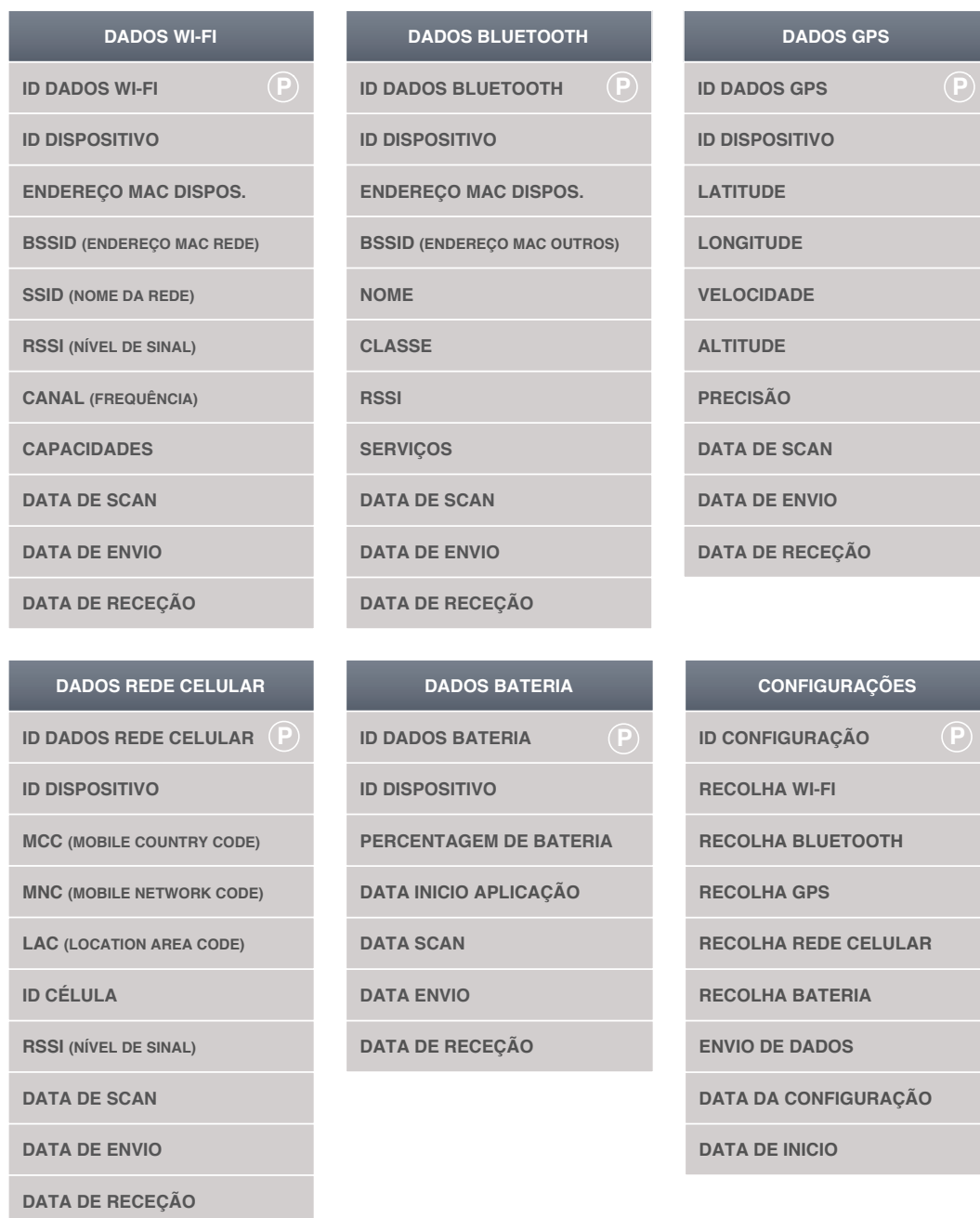


Figura 6.12: Diagrama Entidade Relação (DER) da BD dados recolhidos (Servidor)

2.3 Recolha de Dados dos Interfaces

Cada um dos módulos individuais inicialmente desenvolvidos teve como função a recolha de dados de um dos interfaces. Após desenvolvidos e devidamente testados, através de aplicações que permitiram a visualização dos dados recolhidos, foram colocados em funcionamento conjunto e passaram a designar-se por *Tasks* (Tarefas). Estas tarefas podem ser executadas individualmente ou em conjunto.

2.3.1 Amostragem Constante dos Interfaces

Na fase inicial do processo de implementação, as preocupações dirigiram-se para a implementação da recolha de dados propriamente dita, mais precisamente no modo de utilização das APIs de acesso aos interfaces e na recolha das informações necessárias. Desta forma, procedeu-se ao desenvolvimento dos sub-módulos com base em amostragem constante.

Como foi possível constatar anteriormente através das estimativas dos consumos energéticos, esta abordagem não é eficiente, mas numa primeira fase permitiu não só simplificar o processo de desenvolvimento, evitando o aparecimento de erros externos ao processo de recolha, mas também confirmar os consumos estimados.

Na figura 6.13, está representado um fluxograma que exemplifica como se processa a recolha de dados constante de um interface. Foram de imediato introduzidos mecanismos para verificação da disponibilidade dos interfaces, evitando erros que ocorriam se estes não estivessem acessíveis.

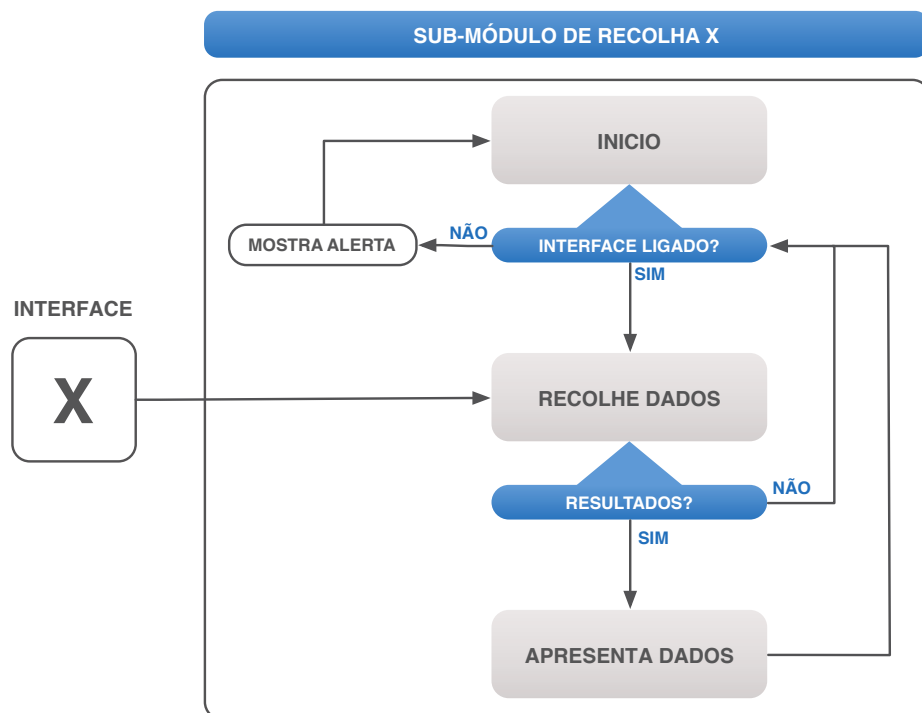


Figura 6.13: Recolha de dados constante de um interface

2.3.2 Tarefas Controladas por Temporizadores

Com objetivo de reduzir o consumo de energia, foi implementado um mecanismo de controle que altera a amostragem de constante para periódica. Este mecanismo baseia-se na implementação de um temporizador (*Timer*) para cada interface. Através da configuração de cada temporizador, é possível definir diferentes períodos de amostragem, reduzindo a utilização dos interfaces e desfasando a execução das várias tarefas. O desfasamento entre as tarefas, permite melhorar a eficiência energética, evitando que as tarefas ocorram em simultâneo, e assim diminuindo o nível de utilização do CPU.

Os temporizadores são configurados segundo três parâmetros:

- Tarefa a executar: Tarefa que deve ser executada sempre que o temporizador termina.
- Tempo de início: Valor em segundos, após o qual deve ser executada pela primeira vez a tarefa.
- Tempo de repetição: Valor em segundos, considerado após a primeira execução para a configuração do temporizador, determinando o intervalo entre as próximas execuções.

O funcionamento geral de um temporizador pode ser observado na figura 6.14. Implementando esta solução, observaram-se melhorias significativas na autonomia da bateria.

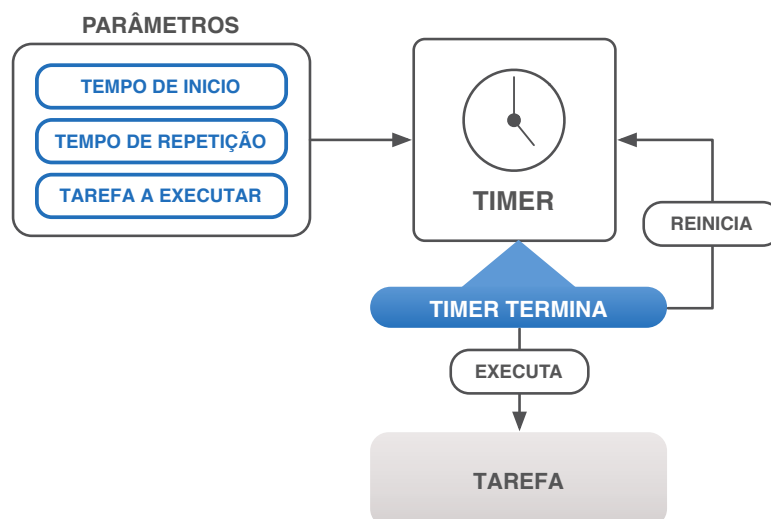


Figura 6.14: Funcionamento dos *Timers* implementados

Como referido anteriormente, as tarefas que são executadas quando os temporizadores terminam, são os sub-módulos de recolha desenvolvidos em primeira instância. Contudo, após a introdução dos temporizadores, o seu funcionamento sofreu ligeiras alterações, sendo executadas periodicamente, as tarefas terminam sempre que realizam a sua função, como apresentado no fluxograma da figura 6.16.

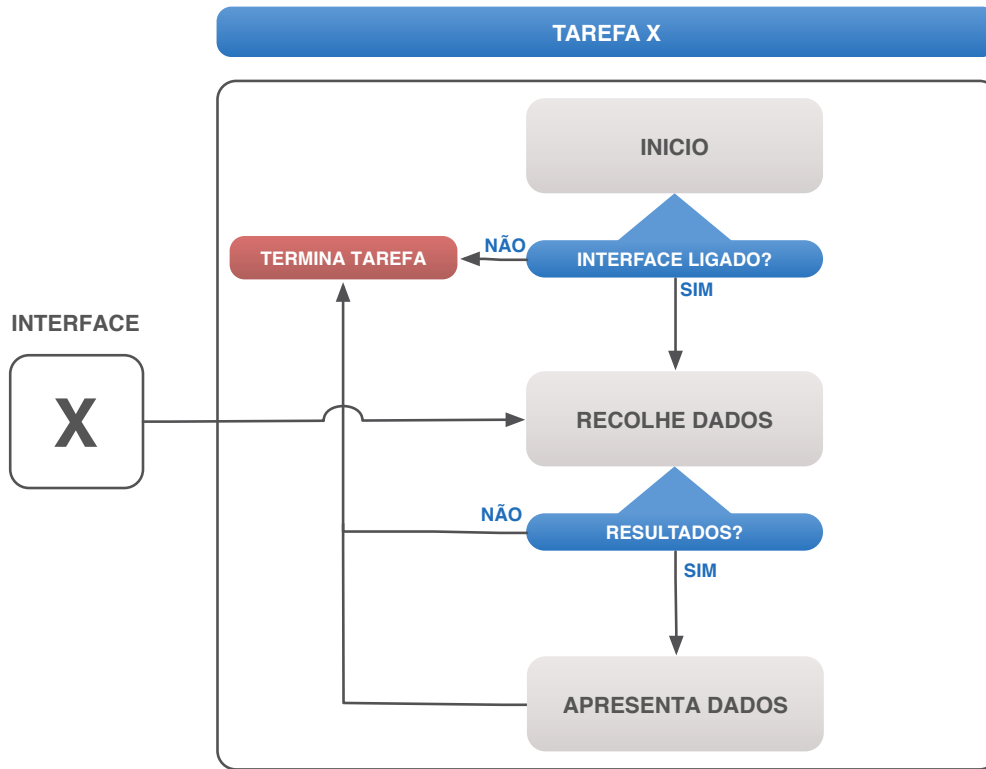


Figura 6.15: Tarefa de recolha e de dados executada por um *Timer*

As tarefas executadas periodicamente para recolha e envio de dados são as seguintes:

- WifiTask: Responsável por recolher informações sobre os pontos de acesso *Wi-Fi*.
- GSMTask: Recolhe informações sobre a rede celular (Estações base atual e vizinhas).
- GPSTask: Permite a recolha da posição GPS atual, entre outras informações.
- BluetoothTask: Responsável por detetar dispositivos *Bluetooth* nas proximidades e recolher informação sobre os mesmos.
- BatteryTask: Permite monitorizar e recolher dados sobre o nível de bateria.
- SendDataTask: Responsável pelo envio dos dados armazenados.

2.4 Armazenamento dos Dados

A ligação constante à Internet nem sempre é possível, por este motivo os dados são armazenados localmente quando são recolhidos. Para o armazenamento local temporário, foi especificada uma base de dados SQLite, apresentada anteriormente (Fig. 6.11), sendo implementada uma classe designada *DatabaseHandler*, que inclui os métodos que permitem a sua utilização. Esta classe fornece ao *DataManager* acesso aos métodos para guardar, ler, e apagar os dados recolhidos. Adicionalmente, permite guardar, atualizar e apagar as configurações de recolha, ou seja os valores dos vários temporizadores (Fig.6.16).

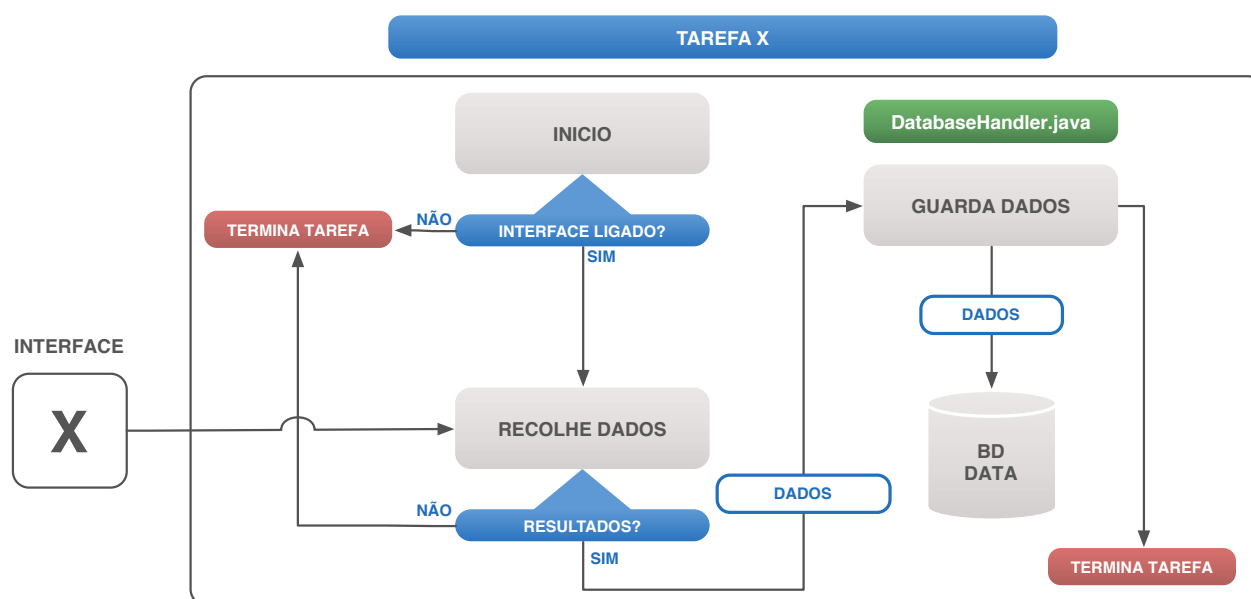


Figura 6.16: Exemplo de uma tarefa de recolha e armazenamento de dados

2.5 Envio dos Dados

Na fase inicial não se distinguiu o tipo de ligação, para o envio dos dados, contudo percebeu-se que isso seria um problema. Diversos utilizadores de *smartphones* não possuem um plano de dados móveis, ou são limitados a um determinado tráfego. Para o envio dos dados adotou-se uma abordagem oportunista. Assim quando é detetada ligação à Internet através de Wi-Fi, é iniciado o envio dos dados armazenados.

Vários fatores, como a utilização de períodos de amostragem curtos, a disponibilidade constante dos interfaces, ou poucas oportunidades de envio dos dados recolhidos, conduzem ao armazenamento de um grande volume de dados, sendo consumido tráfego significativo no seu envio. Esta situação representaria um problema para os utilizadores que utilizam os dados móveis, tendo consequentemente impacto na utilização da aplicação.

Assim, procurando defender os interesses dos utilizadores e da própria aplicação, o envio

dos dados é processado apenas através de redes Wi-Fi.

Para o envio dos dados foi criada uma tarefa específica e independente, executada periodicamente através de um temporizador. Esta tarefa, utiliza os métodos da classe *DatabaseHandler*, para a leitura dos dados armazenados na base de dados, enviando-os para o servidor utilizando a classe *DataSender*. Esta classe, foi desenvolvida não só para o envio dos dados para o servidor, mas também para a troca de informações necessárias ao funcionamento do *DataManager*, como o pedido e receção de configurações. Adicionalmente cria um nível de abstração, sendo responsável por executar os comandos REST.

Após a receção dos dados no servidor, é identificado o interface pelo qual foram recolhidos, sendo armazenados na base de dados MySQL especificada (Fig.6.12).

Na figura 6.17 está representado de forma sintetizada o processo de leitura e envio dados e armazenamento no servidor.

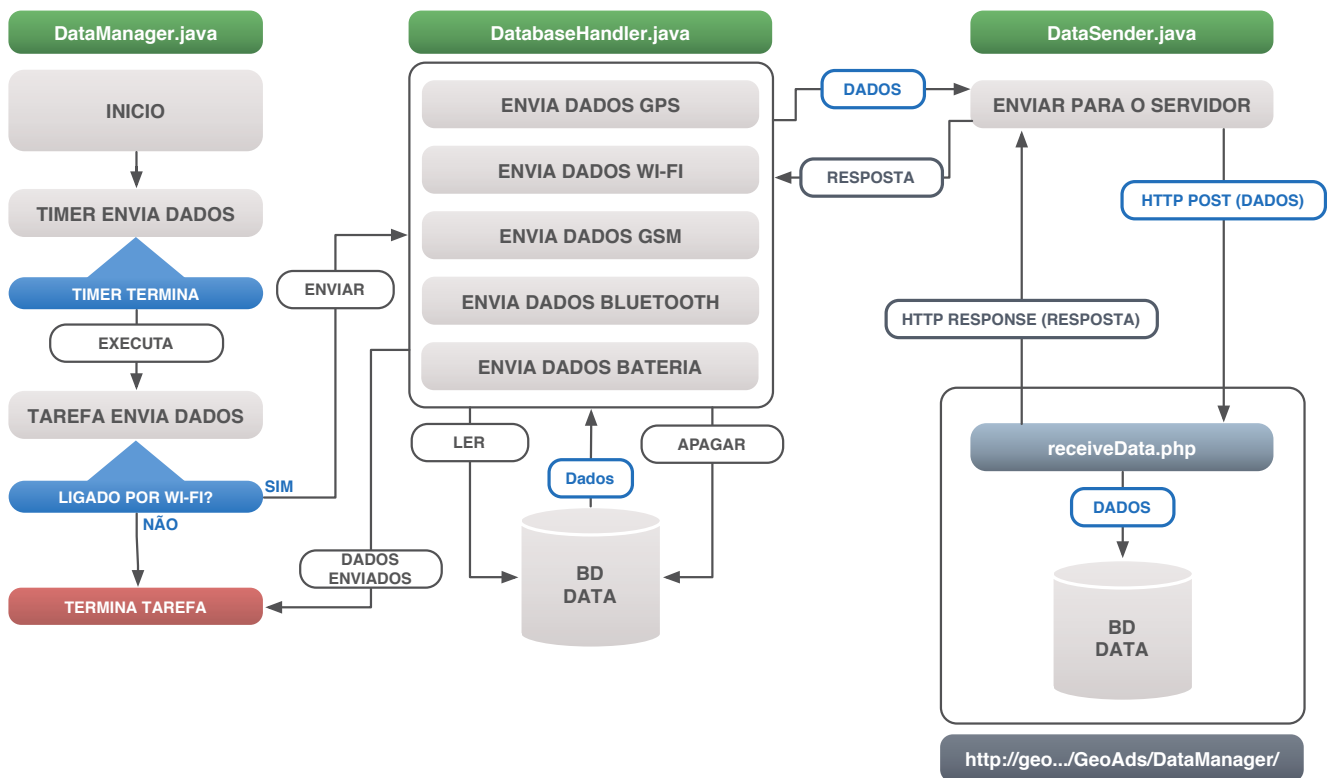


Figura 6.17: Processo de envio de dados armazenados para o servidor

2.5.1 Segurança dos Dados Enviados

Como foi abordado no estado da arte, a maioria das aplicações de *sensing* não implementam mecanismos de segurança, de privacidade e de integridade dos dados. Por consequência, é principalmente posta em causa a privacidade dos utilizadores.

Quando abordada no contexto das plataformas móveis, esta questão tem um impacto muito superior, não podendo de forma alguma ser desprezada. Por outro lado, é importante para o projeto garantir a integridade dos dados, de modo a que não sejam adulterados ou fabricados por terceiros. A solução para este problema reside na implementação de um mecanismo que utiliza criptografia para garantir a proteção dos dados. A criptografia consiste no estudo de princípios e técnicas, pelas quais a informação pode ser transformada da sua forma original para uma forma ilegível, de modo que apenas o recetor autorizado da mensagem tenha acesso à informação original.

Nos dias atuais, em que grande parte dos dados são digitais, o processo de criptografia é basicamente conseguido através de algoritmos que cifram os dados a partir de uma determinada chave ou par de chaves, dependendo do sistema criptográfico adotado. Deste modo, o primeiro passo é a identificação do algoritmo ou do método criptográfico que dê resposta aos desafios.

■ Cifras e Chaves

Uma cifra é um algoritmo criptográfico que consiste numa função matemática injetiva, que transforma texto limpo num criptograma. No passado, os detalhes das cifras eram secretos, sendo nesse princípio que residia a sua segurança. Contudo, na criptografia moderna, não é possível manter a segurança de uma cifra ocultando o seu funcionamento. A qualidade de uma cifra, distingue-se pelo tempo que permanece inquebrável, sendo o seu funcionamento do conhecimento geral.

No que diz respeito às chaves que utiliza, uma cifra pode ser:

- Simétrica: As chaves para cifrar e decifrar a mensagem são iguais. Os interlocutores partilham a mesma chave, que tem de ser previamente definida e mantida secreta.
- Assimétrica: As chaves para cifrar e decifrar são diferentes, sendo uma chave pública e outra privada. A chave privada tem de ser mantida secreta, apenas o recetor da mensagem deve conhece-la. Por exemplo, a entidade A utiliza a chave pública da entidade B para cifrar uma mensagem; ao receber a mensagem a entidade B utiliza a sua chave privada para decifrar. Desta forma, um intruso pode conhecer a chave pública, sem que isso comprometa a segurança da cifra, pois apenas o recetor conhece a chave para decifrar.

■ Solução Implementada

Optou-se pela implementação de um algoritmo baseado em chaves simétricas. Esta solução permite garantir a segurança, a privacidade e a integridade dos dados. Sendo uma solução mais simples e com menor impacto na performance da aplicação do que uma solução baseada em chaves assimétricas.

Desta forma, foi utilizado o algoritmo de encriptação por blocos *Advanced Encryption Standard* (AES), também conhecido por Rijndael. Este algoritmo é utilizado como padrão pelo governo dos Estados Unidos. O tamanho da chave deste algoritmo pode ser de 128, 192, ou 256 bits. A utilização de chaves de 192 e 256 bits é recomendada para informações altamente secretas.

Sendo o AES uma cifra por blocos, pode ser utilizada em diferentes modos de funcionamento. Dependendo do modo de funcionamento, varia a segurança, a eficiência, a ocorrência e tolerância a erros e o *throughput*.

O modo escolhido foi o CBC (*Cipher Block Chaining*), pois permite uma segurança mais elevada. O primeiro bloco é combinado com um vetor de inicialização que pode ser aleatório, sendo que o vetor de inicialização não necessita de ser secreto, mas tem de ser utilizado na decifragem. Os padrões do texto são mascarados pela operação XOR. Mesmo que o texto original seja igual, o criptograma resultante é sempre diferente, o que impede a ocorrência de ataques por *code book* e por repetição.

Em relação ao tamanho da chave utilizou-se AES-128, pois a utilização de AES-256 necessita de 4 iterações adicionais para misturar os restantes bits da chave, tornando o algoritmo mais lento, pois requer mais processamento. A utilização de AES-128 é adequado para o tipo de informação que se pretende proteger.

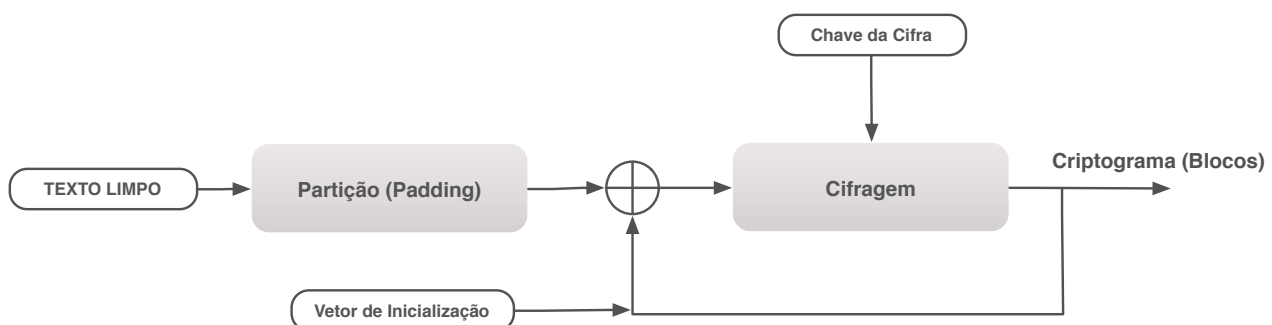


Figura 6.18: Cifragem dos dados AES-128 em modo CBC

2.6 Execução do *DataManager* como Serviço Android

Um dos desafios deste projeto é maximizar a quantidade de dados recolhidos. Aumentando a periodicidade da recolha para cada interface, a quantidade de dados recolhidos iria consequentemente aumentar. No entanto, esta solução não é benéfica, como discutido anteriormente.

A execução do módulo *DataManager* juntamente com a aplicação GeoAnúncios torna-o dependente do ciclo de vida da aplicação, ou seja, o *DataManager* é executado quando o utilizador abre a aplicação e termina quando a aplicação é encerrada (Fig. 6.19). Neste cenário, apenas seriam recolhidos dados enquanto a aplicação estivesse em execução ou minimizada. Assim, a recolha de dados seria ineficiente, visto que a maioria dos utilizadores interage com as aplicações por curtos períodos de tempo, podendo demorar dias até voltar a abrir a aplicação. Acresce a preocupação dos utilizadores em poupar a bateria, que os leva a fechar as aplicações quando não as estão a utilizar. Por este motivo, o módulo *DataManager* foi implementado como um serviço Android. A partir do momento que é executado, continua em execução independente do ciclo de vida da aplicação que o inicia (Fig. 6.20).



Figura 6.19: Módulo GeoAnúncios e módulo *DataManager* juntos numa única aplicação

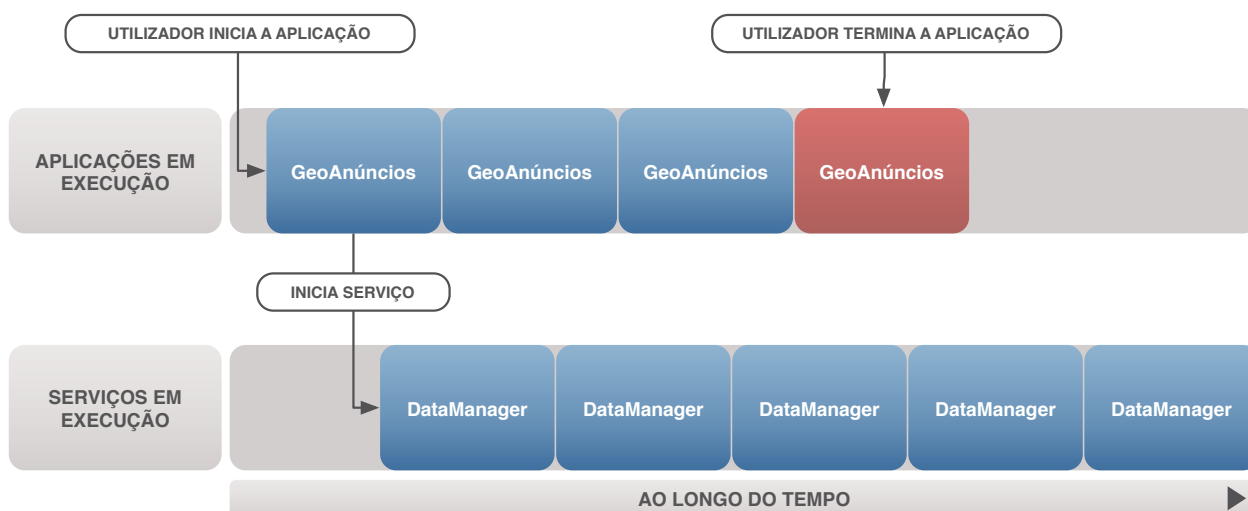


Figura 6.20: Módulo GeoAnúncios como aplicação e módulo *DataManager* como serviço

2.7 Maximizar o Tempo de Execução do *DataManager*

Como explicado anteriormente, o serviço *DataManager* é iniciado juntamente com a aplicação GeoAnúncios e continua em execução de forma independente. No entanto, é dependente da inicialização da aplicação GeoAnúncios, o que pode ser um problema.

Existem diversas situações que podem levar o serviço a ser terminado, como por exemplo, quando o telefone é desligado, se o serviço falhar por algum motivo e terminar ou se o Android devido à requisição de memória para outra aplicação, terminar o serviço para libertar recursos. Nestas situações, o serviço apenas voltaria a ser iniciado quando a aplicação fosse aberta novamente pelo utilizador. Como o nível de utilização da aplicação pode variar entre os utilizadores, em alguns casos o serviço não voltaria a ser executado durante um longo período de tempo.

Para a resolução deste problema, foram implementados dois mecanismos que garantem que o serviço esteja em execução durante o máximo de tempo possível. Utilizando os *BroadcastReceivers* do Android, é possível receber notificações do sistema quando determinados eventos ocorrem. Deste modo, quando o telefone liga, é difundido pelo sistema Android a notificação do evento de *BOOT*. Se estiver registado no manifesto da aplicação um *receiver* para este evento, é automaticamente executado um método definido. Neste método é iniciado o serviço *DataManager*. O evento de *BOOT*, endereça a situação do telefone ser desligado.

No caso do serviço ser terminado por um dos restantes motivos, foi registado um *receiver* para a notificação de quando o cabo de energia é desligado. Assim, se o serviço não estiver em execução, é iniciado nessa altura.

Existem outros eventos do sistema que poderiam ter sido utilizados, como por exemplo quando o cabo de energia é ligado ao *smartphone*. No entanto, o evento de ligação do cabo de energia não seria vantajoso. Quando o utilizador liga o telefone à corrente elétrica não existe movimento, mas quando o utilizador desliga o cabo de energia normalmente pega no telefone e inicia movimento fazendo sentido nessa altura o serviço entrar em funcionamento.

2.8 Reconfiguração Periódica Automática

Para alterar os períodos de amostragem, o administrador do sistema cria um novo conjunto de configurações no servidor. Como as tarefas de recolha de dados são configuradas quando o serviço é iniciado, as configurações iniciais seriam mantidas até que o serviço fosse terminado e iniciado novamente. Isto constituiria um problema, visto que em situações normais o serviço iria estar em execução contínua durante dias ou semanas. Para resolver esta questão, foi adicionado ao *DataManager* um novo temporizador, que verifica se existem novas configurações no servidor periodicamente, caso existam reinicia todas as tarefas utilizando os novos períodos para os temporizadores.

2.9 Controlo da Recolha de Dados

Como discutido anteriormente, quando o *smartphone* está parado os dados recolhidos têm menor utilidade, sendo desperdiçada energia que compromete a autonomia do dispositivo.

A utilização do acelerómetro para a deteção de movimento permite ultrapassar este problema, apresentando consumos reduzidos.

A deteção de movimento através do acelerómetro é relativamente simples, o grande desafio é identificar se o movimento detetado é significativo para considerar que o utilizador está em movimento e assim iniciar a recolha de dados.

Outro desafio está relacionado com o comportamento do sistema Android, quando os *smartphones* estão em modo de *standby*. Neste modo, o CPU fica inativo e todas os processos em execução são suspensos. Esta característica torna impossível executar qualquer função ou utilizar os recursos, os interfaces e os sensores do *smartphone* durante esse período. Consequentemente, não é possível recolher dados durante um longo período de tempo.

2.9.1 Deteção de Movimento

Para os objetivos do presente projeto não é relevante distinguir ou contextualizar o tipo de movimento do utilizador, se ele está a caminhar, a correr ou a viajar de carro, mas sim se existe ou não movimento.

Desta forma, pode-se considerar que o objetivo é detetar a ausência de movimento, para que nesses períodos a recolha de dados seja suspensa ou reduzida.

Os acelerómetros são sensíveis à diferença entre a aceleração linear do sensor e do campo gravitacional. Quando é realizada a leitura do acelerómetro, estão disponíveis três valores, correspondentes aos valores da aceleração nos três eixos (X, Y e Z), como mostra a figura 6.21.

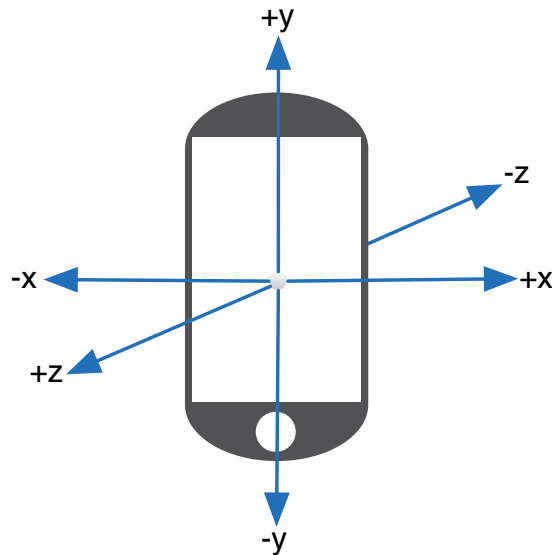


Figura 6.21: Eixos de um acelerômetro

Os valores obtidos em X, Y e Z, através da API do Android para o acesso ao acelerômetro, representam a aceleração em cada eixo, em m/s^2 . Os valores podem ser positivos ou negativos, dependendo da posição e do movimento do telefone. Idealmente e na teoria, se o telefone estiver colocado numa superfície nivelada, os valores lidos devem ser zero em X e Y e -9.81^7 em Z. Se o telefone estiver assente sobre o seu lado esquerdo, devem ser lidos $X = -9.81$ e Y e Z zero, assim sucessivamente.

Na prática, o acelerômetro é muito sensível, o mais pequeno movimento produz uma reação instantânea. Devido à natureza do hardware do sensor e à física, mesmo que o telefone esteja completamente estático, observa-se um nível constante de ruído, resultando em ligeiras alterações dos valores.

■ Algoritmo de Detecção

Numa abordagem simplista, é possível detetar a ausência de movimento através da leitura dos valores da aceleração dos três eixos e com esses valores calcular a magnitude da aceleração linear. Se esta magnitude for inferior a um determinado valor (por exemplo 0.3^8), considera-se que não existe movimento. A magnitude da aceleração linear é um único valor positivo, que representa a aceleração linear nos três eixos. No entanto, esta abordagem não é eficiente, pois como referido o acelerômetro é extremamente sensível, sendo que a mínima alteração da posição do *smartphone* seria considerada movimento.

Por este motivo, revelou-se necessário desenvolver um algoritmo mais complexo, para minimizar este efeito (Fig. 6.22).

⁷ 9.81 m/s^2 é o valor aproximado da aceleração da gravidade.

⁸Mesmo quando o dispositivo está parado, o valor da aceleração linear não é exatamente zero.

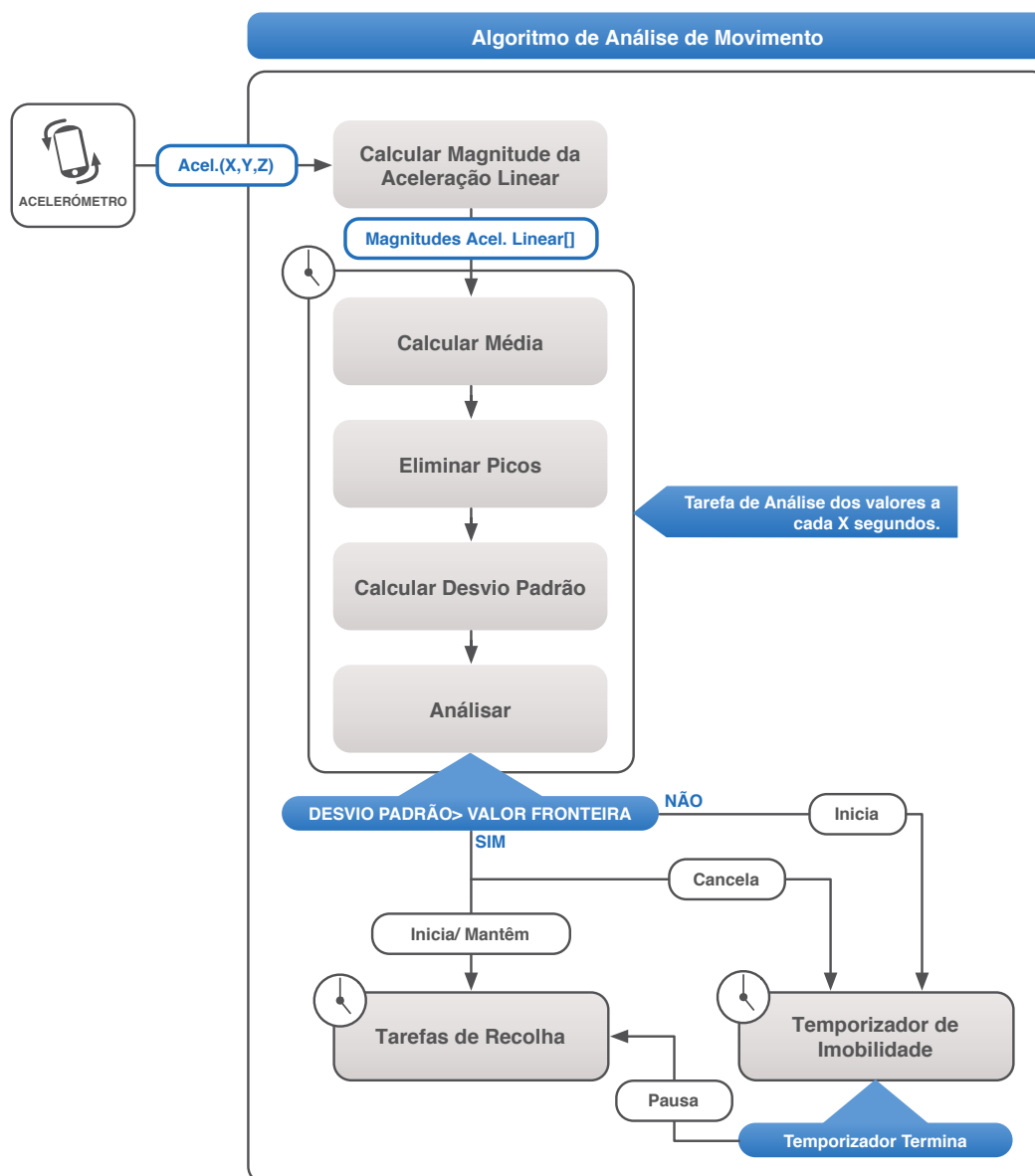


Figura 6.22: Funcionamento do algoritmo de detecção de movimento

O algoritmo desenvolvido para detetar o movimento, considera o valor do desvio padrão da magnitude da aceleração linear das amostras recolhidas durante um determinado período de tempo (por exemplo 10 segundos). Caso não seja detetado qualquer movimento durante um determinado período de tempo (por exemplo 2 minutos), as tarefas de recolha são suspensas. Quando é detetado movimento, as tarefas de recolha são iniciadas ou mantidas se já estiverem ativas.

Para reduzir o ruído da amostragem, é utilizado um filtro de Kalman no cálculo da magnitude da aceleração linear. O filtro considera não só a amostra atual mas também a anterior, dando assim origem a valores mais precisos do que aqueles que seriam obtidos baseados numa única leitura.

Para definir o valor do desvio padrão, a utilizar no algoritmo como fronteira de distinção entre os estados do dispositivo (em movimento e estático), foi necessário analisar os valores recolhidos através do acelerómetro.

O gráfico da figura 6.23, representa o movimento de um utilizador a caminhar. A azul estão representadas as amostras da magnitude da aceleração linear ao longo do tempo e a vermelho, a média das amostras recolhidas a cada 10 segundos.

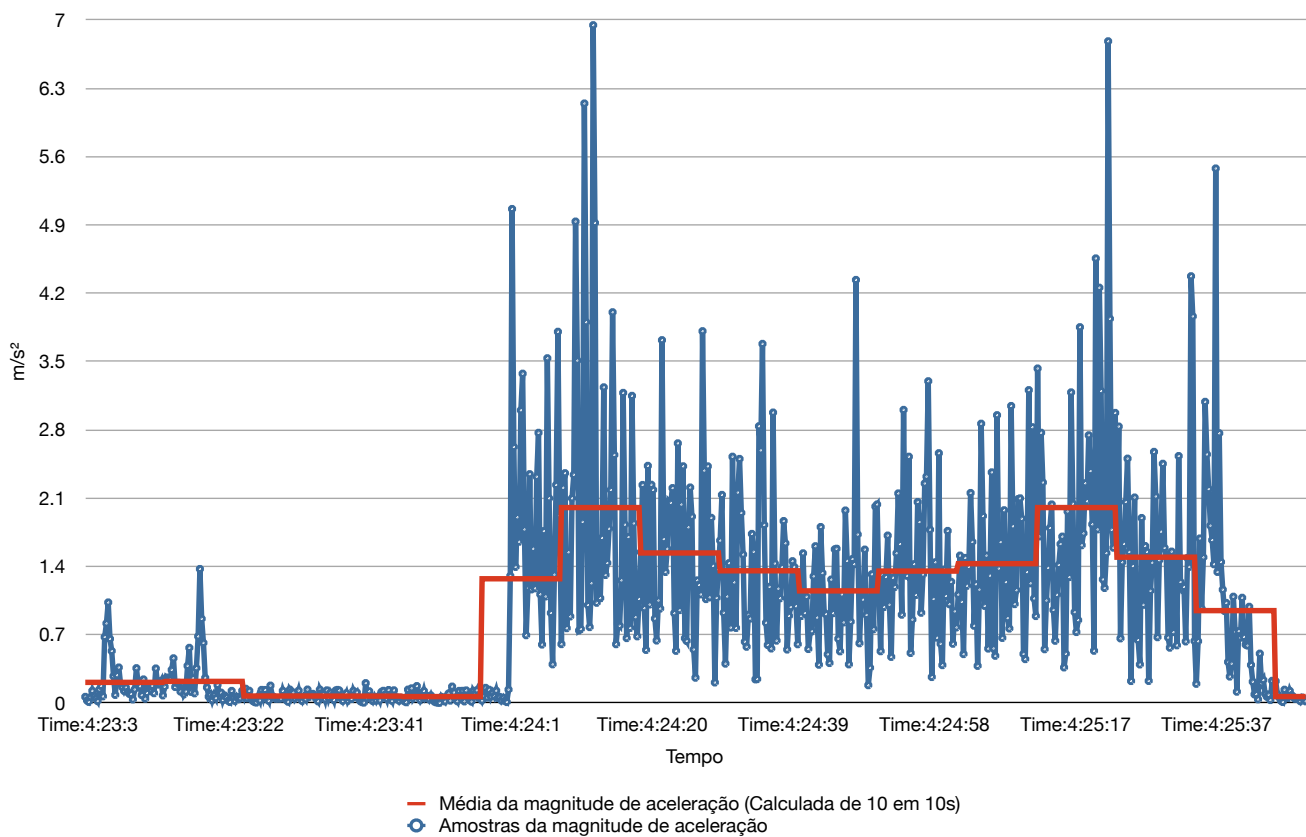


Figura 6.23: Valores da magnitude da aceleração linear do acelerómetro a caminhar

Como é possível observar, existe um período de pouco movimento, no qual os valores são praticamente nulos, sendo posteriormente iniciado o movimento de forma regular. Este gráfico demonstra que durante a amostragem, existem diversos picos (valores muito superiores à média), mesmo sendo utilizado o filtro de Kalman.

De modo a que o cálculo do desvio padrão seja mais preciso, foi necessário eliminar estes picos. As amostras que apresentam uma magnitude significativamente diferente da média (por exemplo $\pm 1.2 \text{ m/s}^2$) são excluídas. O gráfico da figura 6.24 representa o período em que existe movimento, sem os valores considerados como picos.

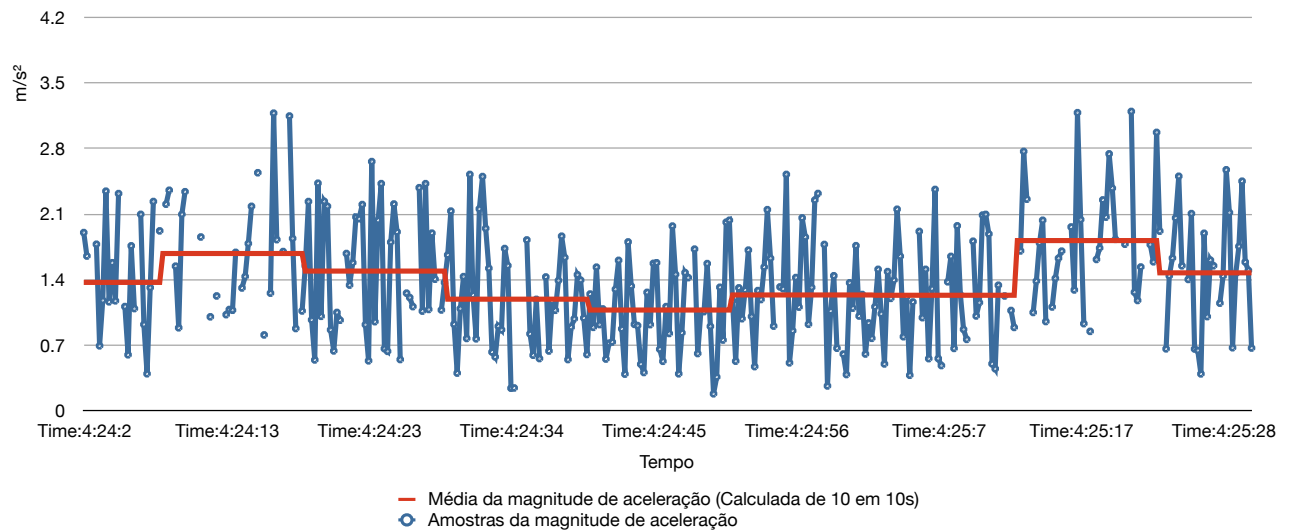


Figura 6.24: Valores da magnitude da aceleração linear com redução dos picos

Através do gráfico, verifica-se que o resultado é mais linear, sendo assim possível calcular de forma mais precisa o desvio padrão que representa o movimento do utilizador.

Após algumas medições, concluiu-se que em média, o desvio padrão da magnitude de aceleração gerada por um utilizador a caminhar com o seu *smartphone* no bolso, é de $\approx 0.43 \text{ m/s}^2$, valor que vai de encontro a resultados obtidos em outros estudos[CC11]. Desta forma se o desvio padrão for igual ou superior a este valor, considera-se que o utilizador está em movimento, iniciando-se assim a recolha de dados.

O algoritmo desenvolvido mostrou-se eficiente, no entanto com a aplicação deste valor como fronteira, apenas seria considerado movimento se o utilizador se deslocasse a pé. No caso do utilizador viajar de carro, os valores da aceleração detetados pelo acelerómetro são consideravelmente inferiores, porque as alterações na posição do telefone são menos significativas, gerando assim menor variação dos valores dos eixos do acelerómetro. Neste caso, o algoritmo definia o estado do telefone como parado. Sendo o objetivo a deteção da ausência de movimento, esta calibração iria gerar falsos positivos.

Os gráficos 6.25 e 6.26, representam a magnitude da aceleração linear aquando da deslocação de carro, com o telefone pousado no carro e com o telefone no bolso, respetivamente.

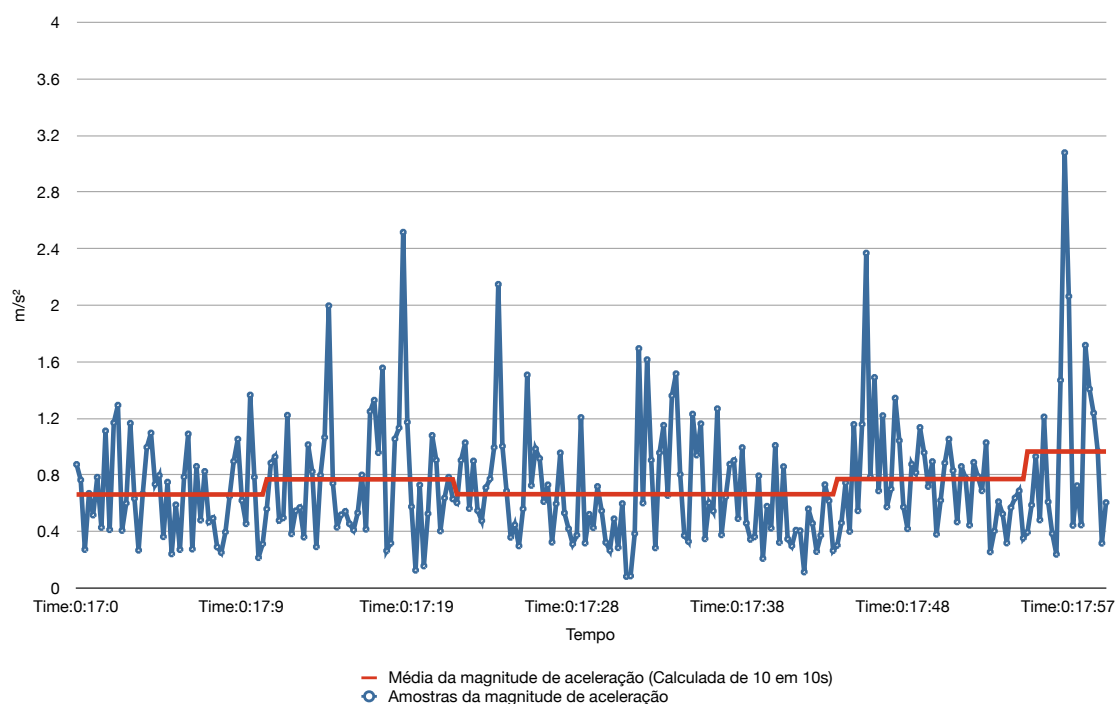


Figura 6.25: Valores da magnitude da aceleração linear do acelerómetro de carro (telefone pousado)

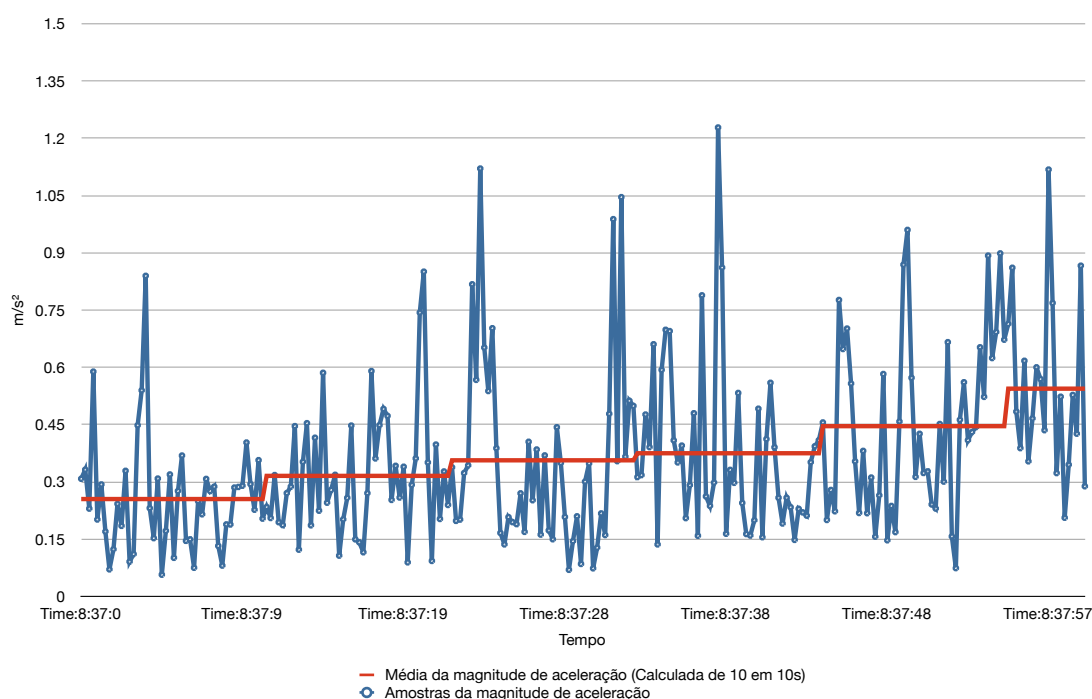


Figura 6.26: Valores da magnitude da aceleração linear do acelerómetro de carro (telefone no bolso)

Como é possível observar através dos gráficos, a magnitude dos valores de aceleração com o telefone pousado no carro é superior à magnitude detetada quando no bolso. No segundo caso nota-se que existe uma maior irregularidade nos valores, isto deve-se aos movimentos das pernas do condutor durante a condução.

Após a eliminação dos picos, os valores do desvio padrão são de $\approx 0.35 \text{ m/s}^2$ quando o

telefone está pousado no automóvel e de $\approx 0.22 \text{ m/s}^2$ quando utilizador tem o telefone no bolso. Desta forma, considerou-se o valor mais baixo como valor fronteira do algoritmo, de modo a detetar ambas as situações.

■ Manter a Recolha Ativa Quando o *Smartphone* é Bloqueado

Como abordado anteriormente, quando o *smartphone* é bloqueado pelo utilizador ou entra automaticamente em modo inativo, o sistema Android suspende todas as tarefas que estão a ser executadas pelo CPU, até que o *smartphone* volte a ser utilizado. Esta característica tem como objetivo poupar energia quando o dispositivo não está a ser utilizado. No entanto, para os objetivos deste projeto é um problema crítico, pois interrompe as tarefas de recolha de dados.

A solução está na utilização do *WakeLock* da classe *PowerManager* do Android. O *WakeLock* permite que uma aplicação mantenha o controlo sobre o estado do dispositivo, podendo manter os recursos ativos mesmo quando o telefone não está a ser utilizado. Existem diferentes tipos de *WakeLock*, desde o completo que mantém ativos o ecrã, o teclado e o CPU, ao parcial que mantém apenas o CPU ativo. Para os objetivos deste projeto é necessário manter apenas o CPU ativo, sendo por isso o modo parcial o mais indicado.

Porém, a utilização de *WakeLocks* implica elevados consumos de energia. A própria documentação do Android alerta os *developers* para este facto. O consumo depende do tipo de *WakeLock* utilizado e do tempo durante o qual é utilizado. Assim, é recomendado que o *WakeLock* seja utilizado da seguinte forma:

1. Adquirir o *WakeLock*;
2. Executar Tarefas;
3. Libertar *WakeLock*;

No contexto deste projeto, as tarefas de recolha de dados são periódicas e devem ser executadas sempre que existe movimento. Se o *WakeLock* for libertado depois da primeira execução o CPU adormece, não sendo possível executar as tarefas seguintes. Se por outro lado o *WakeLock* não for libertado, o constante funcionamento do CPU terá um grande impacto na autonomia do dispositivo, isto porque consome energia constantemente e os outros processos aproveitam o facto do CPU estar ativo para executarem as suas tarefas.

De modo a minimizar este problema, foi utilizada a deteção de movimento aliada a um alarme do sistema. Se não for detetado movimento durante um determinado período de tempo (por exemplo 2 minutos), além da recolha de dados ser suspensa, o *WakeLock* é libertado

e o CPU adormece. No entanto, quando o CPU está adormecido, não é possível detetar o movimento para voltar a iniciar a recolha de dados.

Por este motivo, foi implementado um mecanismo adicional, que utiliza a classe *Alarm Manager* do Android. Esta classe permite o acesso ao serviço de alarmes do sistema, sendo possível agendar a execução de um determinado método da aplicação. Quando o alarme termina, o *Intent* que foi registado para o evento é difundido pelo sistema. Se o serviço não estiver em execução, é iniciado e executado o método definido como recetor do alarme.

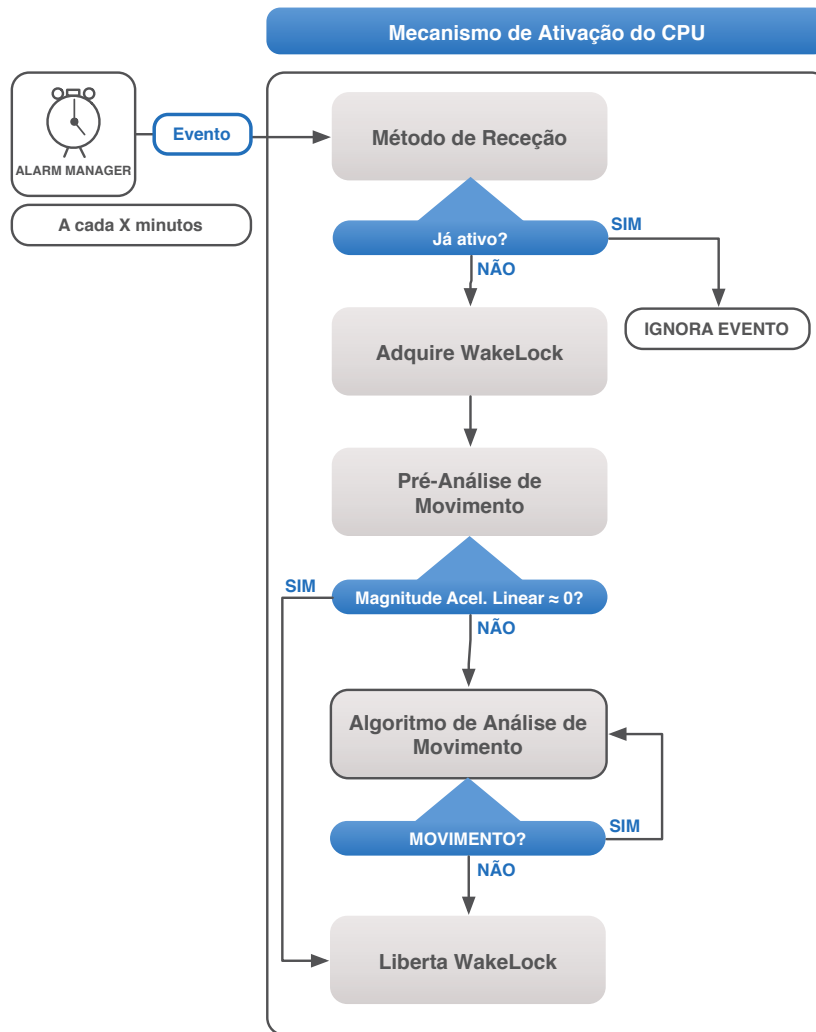


Figura 6.27: Mecanismo de ativação do CPU

Através da utilização do alarme, é possível acordar o CPU periodicamente (por exemplo a cada 2 minutos), executando o método de receção. Neste método, é implementado um *WakeLock* para manter o CPU ativo, sendo de seguida realizada uma pré-análise dos valores do acelerómetro. Caso não seja detetado qualquer movimento, o *WakeLock* é libertado imediatamente e o CPU adormece. Se for detetado movimento na pré-análise, é aplicado o algoritmo de análise de movimento. Se o valor fronteira do algoritmo for ultrapassado, o CPU é mantido

ativo, sendo iniciada a recolha de dados. Posteriormente, quando o algoritmo de análise não detetar de novo movimento, o *WakeLock* é libertado e o CPU adormece, até que o próximo alarme seja difundido.

3 GeoAnúncios

Ao longo desta secção, é descrita a implementação da aplicação e do serviço GeoAnúncios. São explicadas as funcionalidades implementadas, assim como a estrutura da aplicação.

3.1 Base de Dados

No sistema GeoAnúncios existem duas bases de dados. A base de dados central no servidor, onde são armazenados todos os conteúdos do serviço, e uma base de dados local na aplicação móvel, que permite a disponibilização de algumas funcionalidades sem ligação à Internet. Nas figuras 6.28 e 6.29, estão representados os diagramas entidade relação (DER) das bases de dados especificadas para a aplicação móvel e para o servidor, respetivamente.

■ Aplicação Móvel

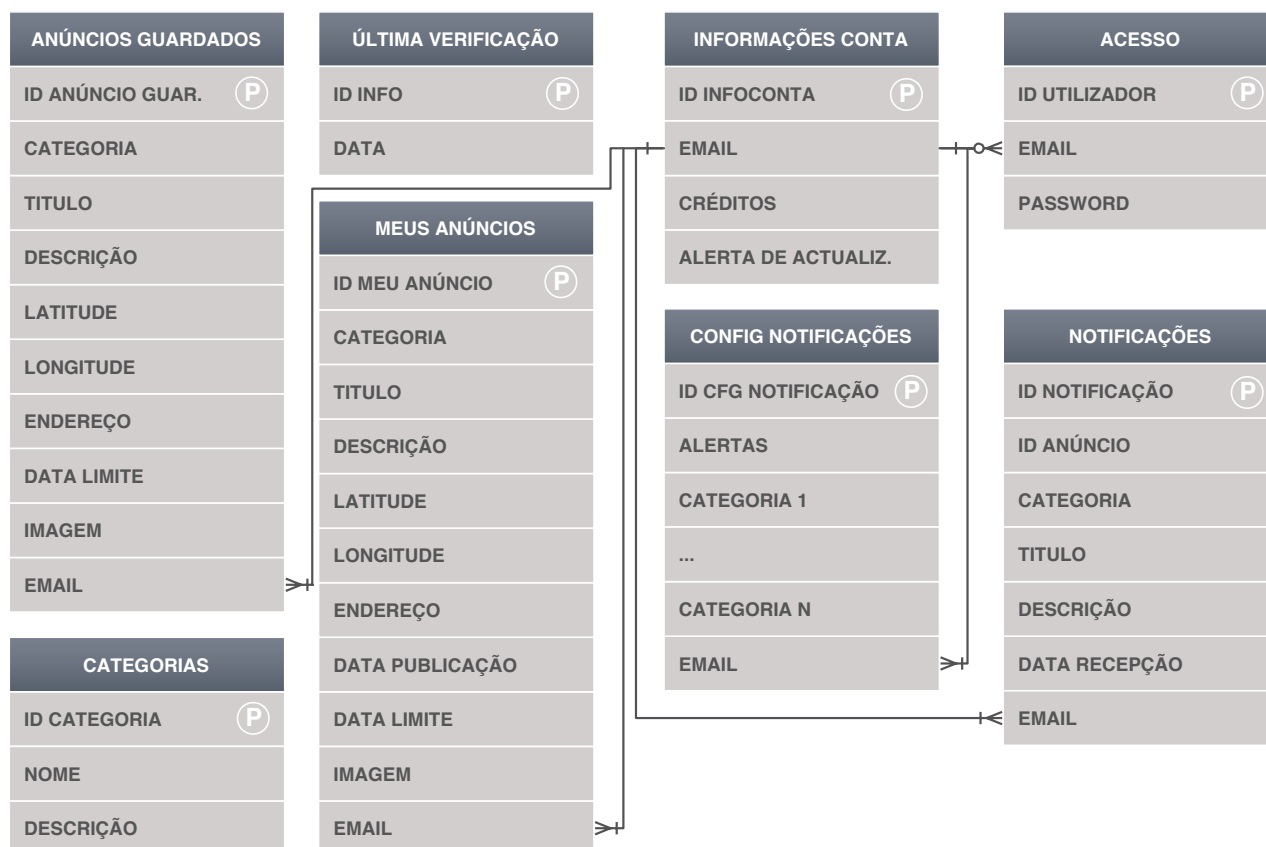


Figura 6.28: DER da BD GeoAnúncios *Cache* (Aplicação Móvel)

Para tornar mais clara a função de algumas tabelas:

- **Anúncios guardados:** Anúncios criados pelo utilizador mas ainda por publicar.
- **Meus anúncios:** Anúncios já publicados pelo utilizador.
- **Última verificação:** Permite guardar a data e hora da última verificação de novos conteúdos no servidor. Permite ao servidor determinar os conteúdos de que o utilizador já foi notificado.
- **Acesso:** Armazenamento dos dados no primeiro acesso, permitindo o *login* automático quando o utilizador abrir novamente a aplicação.
- **Config. Notificações:** Configurações do utilizador para a receção de notificações, podendo desativar os alertas e escolher as categorias a receber notificações.

■ Servidor

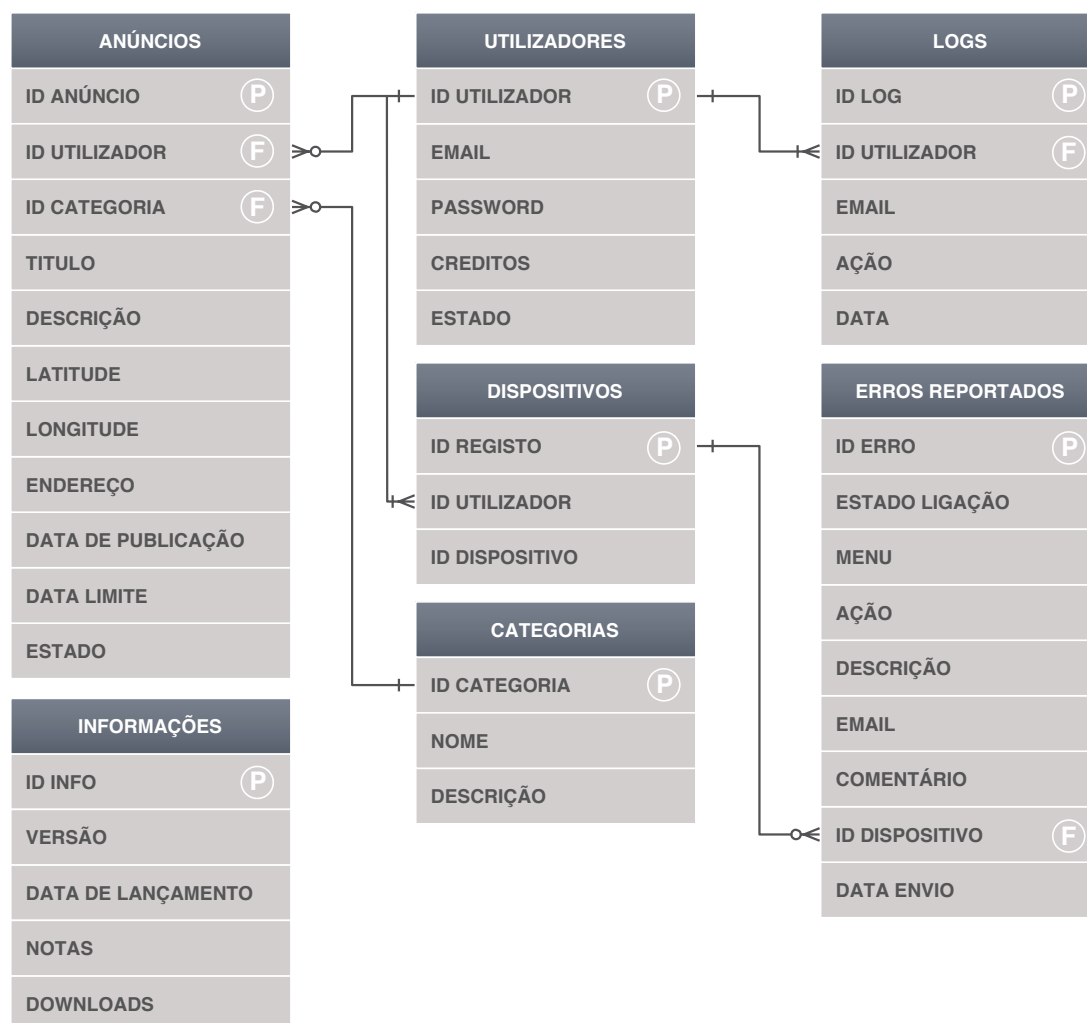


Figura 6.29: Diagrama Entidade Relação (DER) da BD GeoAnúncios (Servidor)

3.2 Registo, *Login* e Créditos

Após a instalação da aplicação, os utilizadores necessitam de se registar e efetuar o *login* (Fig. 6.30) para acederem às funcionalidades da aplicação. Como foi discutido anteriormente quando se concebeu a aplicação, o registo permite entre outras funções controlar o acesso ao sistema, permitindo que os administradores possam bloquear o acesso em caso de abuso do serviço e simplificar o processo de gestão do serviço GeoAnúncios.

Figura 6.30: Menu inicial da aplicação GeoAnúncios

■ Registo

O registo é feito através da aplicação, sendo necessário a introdução de um email e de uma password.

O utilizador tem de introduzir duas vezes cada campo, para reduzir a ocorrência de registos incorretos. O email introduzido é validado, verificando se o padrão do email é correto (texto@entidade.dominio), e se não foram introduzidos caracteres inválidos. Se os dados forem válidos, são enviados para o servidor, que verifica se o email introduzido já se encontra registado. Se ainda não estiver o registo é efetuado.

A password do utilizador não é armazenada na base de dados em texto legível por questões de privacidade, sendo por isso calculado o *Hash* da mesma. Se dois utilizadores definirem a mesma password, o *Hash* é o mesmo. Para que esta situação não aconteça, adiciona-se à

password original uma *String* aleatória, chamada *Salt*, sendo depois calculado o *Hash* dessa combinação. O *Salt* não necessita de ser secreto.

Com este método, ataques por *lookup tables*, *reverse lookup tables* e *rainbow tables* tornam-se ineficientes. Um atacante não sabe antecipadamente qual será o *Salt* utilizado, assim não pode construir as tabelas para o ataque.

O *Hash* é suficiente para que mais tarde seja possível a comparação com o *Hash* da password inserida no *login*, concluindo a autenticação do utilizador.

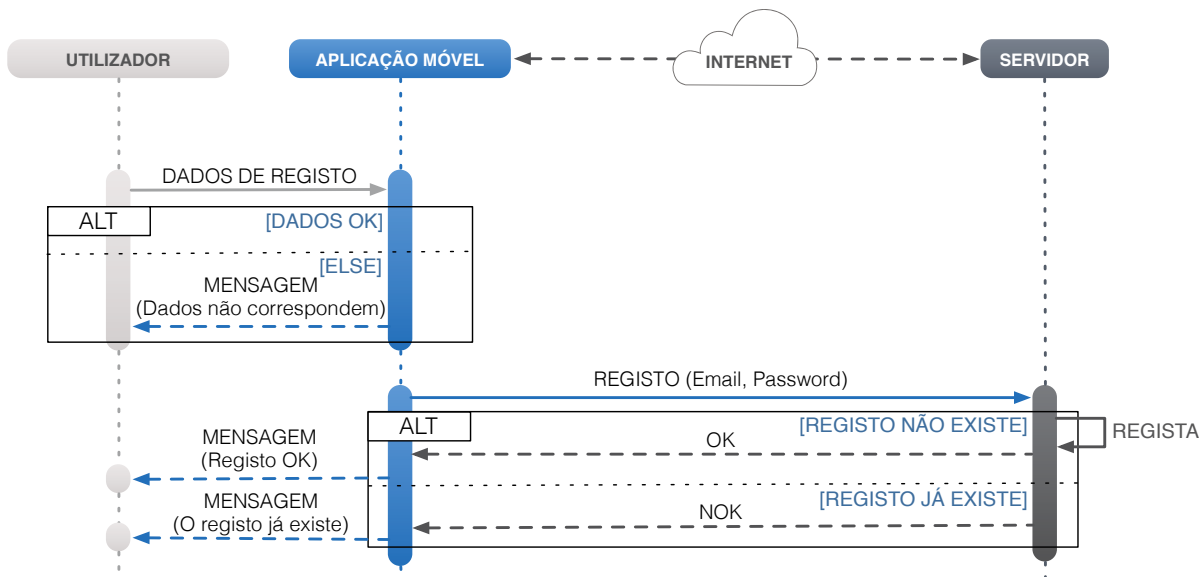


Figura 6.31: Registo de um utilizador

■ Login

Quando o utilizador abre a aplicação, é verificado se existem dados de acesso guardados na base de dados, caso existam a aplicação tentará efetuar automaticamente o *login* com esses dados. No caso de não existirem dados de acesso armazenados, é apresentado o menu de *login*. Depois de introduzir os seus dados de acesso, o utilizador pode escolher guardar-los para que na próxima vez o *login* seja automático.

Após a autenticação dos dados com o servidor, é verificado se o *smartphone* já está registado no sistema, isto porque cada dispositivo não pode estar associado a mais de uma conta de utilizador em simultâneo, para que seja possível atribuir os créditos pelos dados recolhidos. Se o dispositivo não estiver registado, é apresentado o menu principal ao utilizador, caso o dispositivo já esteja registado é apresentada uma mensagem informando que aquele dispositivo já está associado a outra conta, dando a opção de cancelar o *login* ou de alterar a associação para a conta atual.

Quando é feito um pedido de *login*, o servidor verifica se a conta de utilizador existe e se não se encontra bloqueada pelo administrador. Se os dados de acesso estiverem corretos é enviada a confirmação de sucesso para a aplicação.

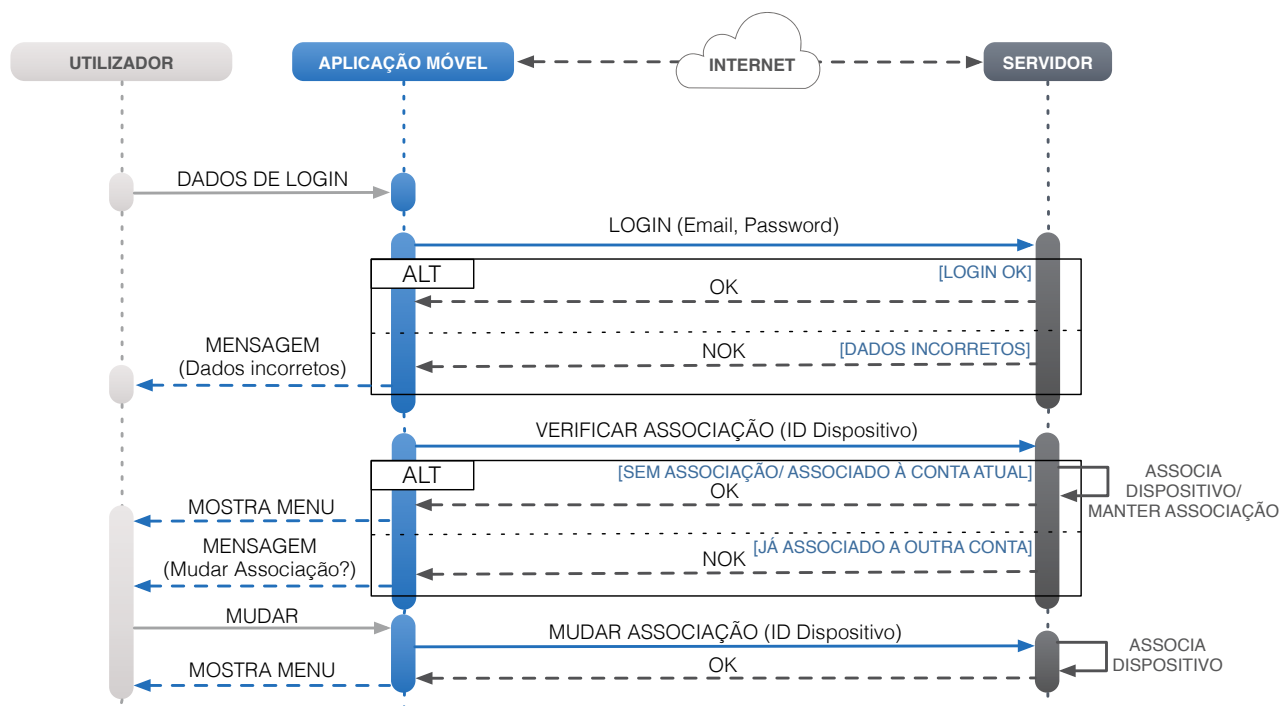


Figura 6.32: *Login* manual de um utilizador

■ Créditos

Os créditos foram implementados como recompensa para os utilizadores que mais contribuem com dados de movimento. Quando são recebidos dados de movimento, é verificado qual o ID do dispositivo que enviou os dados, sendo atribuídos créditos à conta de utilizador à qual o dispositivo está associado.

Como foi mencionado anteriormente, não se pretende a associação do utilizador aos seus dados de movimento. Deste modo, são utilizadas *queries* SQL para atribuir de forma dinâmica os créditos ao utilizador na base de dados do serviço GeoAnúncios através do ID do dispositivo não sendo necessário o conhecimento de dados do utilizador, como por exemplo o email.

3.2.1 Modo *Offline*

Muitos utilizadores não têm acesso constante à Internet no seu dispositivo. Desta forma, para flexibilizar a utilização da aplicação e melhorar a experiência de utilização, foi criado o modo de funcionamento *offline*. Este modo de funcionamento permite que o utilizador tenha acesso às notificações, aos seus anúncios publicados e guardados, e aos seus favoritos. Permite ainda a criação de anúncios, porém para completar a publicação é necessária conexão à Internet.

Para que o utilizador aceda a estas funcionalidades, sem ligação à Internet, deverá ter efetuado previamente o *login* e ter selecionado a opção de guardar os dados de acesso.

São utilizadas bases de dados locais para armazenar as informações necessárias, nomeadamente os anúncios completos que se enquadram nas funcionalidades referidas. Se os anúncios tiverem imagens associadas, são guardadas no cartão de memória do *smartphone*. Quando existe ligação à Internet, a base de dados é sincronizada. Assim, se o utilizador tiver mais do que um dispositivo a informação está disponível em todos os dispositivos.

3.3 Funcionalidades

Quando o utilizador faz a autenticação, tem acesso ao menu principal da aplicação. Neste menu são apresentadas ao utilizador um conjunto de opções (Fig. 6.33).



Figura 6.33: Menu principal da aplicação GeoAnúncios

3.3.1 Opções e Informações

■ Opções

Neste menu o utilizador encontra três funcionalidades. Pode alterar a sua password, pode verificar manualmente se existe uma versão mais recente da aplicação e pode efetuar o *logout* da sua conta de utilizador. Neste menu são apresentados os créditos acumulados e o número de anúncios que pode publicar.

A verificação da existência de atualizações é realizada automaticamente pela aplicação, sendo o utilizador notificado uma única vez por cada nova versão lançada. Por este motivo,

caso o utilizador opte por não atualizar a aplicação no momento que recebe a notificação, poderá posteriormente fazer a atualização através deste menu ou do Google Play.

Quando o utilizador optar por efetuar o *logout*, é informado que deixará de poder aceder à aplicação em modo *offline*, podendo apenas voltar a aceder se existir conectividade para autenticar os dados inseridos.

■ Informações

No menu das informações é apresentado o âmbito do desenvolvimento da aplicação, possibilitando acesso direto ao site para mais informações.

3.3.2 Notificações

As notificações são uma forma de os utilizadores serem informados assim que um novo anúncio do seu interesse seja publicado, sendo recebidas mesmo quando a aplicação não está em execução.

O processo de envio de notificações é baseado em verificações periódicas. Estas verificações, consistem no envio das categorias que o utilizador escolheu para ser notificado e da data e hora da última verificação para o servidor. O servidor, através da categoria e da data de publicação dos anúncios, determina os anúncios que foram publicados depois da última verificação e que se enquadram nas categorias que o utilizador definiu, enviando essas informações para a aplicação.

As novas notificações surgem na área de notificações do Android, desta forma o utilizador é informado sem necessitar de abrir a aplicação. A notificação é identificada na área de notificações, através do ícone dos GeoAnúncios, do título e da categoria do anúncio. O utilizador pode selecionar a notificação, sendo de imediato aberto o menu de notificações da aplicação GeoAnúncios. Neste menu, o utilizador encontra todas as notificações recebidas que ainda não foram vistas; selecionando uma notificação no menu das notificações é aberto o anúncio correspondente. Após ser visualizada, a notificação desaparece automaticamente da lista. Caso seja uma notificação de uma nova versão da aplicação, está disponível a opção para atualizar de imediato. Está também disponível no menu das notificações a opção para limpar todas as notificações recebidas.

3.3.3 Ver Anúncios

Todos os anúncios publicados podem ser encontrados neste menu. Quando o utilizador selecionar a opção ver anúncios, são-lhe apresentados de imediato os anúncios publicados recentemente.

Estão disponíveis dois mecanismos para que o utilizador encontre anúncios que se enquadram nos seus interesses: pode filtrar por categorias, procurar por cidade e/ ou por conteúdo do anúncio introduzindo os termos que deseja procurar.

A pesquisa é realizada enviando as opções de pesquisa e os termos a pesquisar para o servidor. No servidor, estas informações são utilizadas para obter da base de dados os anúncios que satisfazem os parâmetros de pesquisa, sendo enviados para a aplicação. Ao selecionar um dos anúncios que aparecem como resultado da sua pesquisa, é apresentado o seu conteúdo ao utilizador, que pode utilizar todas as funções disponíveis, podendo voltar posteriormente aos resultados.

3.3.4 Ver no Mapa e Perto de Mim

O utilizador pode visualizar os anúncios através do mapa. Esta forma de visualização é dinâmica e intuitiva, proporcionando ao utilizador uma visão geral dos anúncios disponíveis no país. Este modo de visualização é uma das vantagens dos GeoAnúncios, apenas sendo possível, graças à georreferência dos anúncios,

Cada anúncio é representado no mapa pelo logótipo dos GeoAnúncios. O utilizador pode ampliar o mapa, de modo a visualizar melhor os anúncios disponíveis nessa área. Também aqui tem disponível a função de filtragem, que permite apresentar no mapa apenas anúncios de uma categoria. Ao selecionar um dos anúncios, é aberta uma pré-visualização do anúncio. Contendo o ícone que identifica a categoria, o título e a descrição do anúncio, dando ao utilizador a opção de ver os detalhes do anúncio, permitindo visualizar o anúncio completo, a opção de navegar para o local do anúncio e a opção de fechar e continuar a ver os restantes anúncios.

No menu "Perto de Mim", o utilizador encontra as mesmas opções que estão disponíveis no menu Ver no Mapa, mas neste caso a aplicação irá mostrar os anúncios perto da posição atual do utilizador. Caso este se desloque, por exemplo, numa cidade, o mapa irá acompanhar o seu movimento apresentando os anúncios nas imediações.

3.3.5 Navegar para os Anúncios

Como foi salientado anteriormente, a georreferência dos anúncios é uma grande vantagem, possibilitando a implementação de funcionalidades interessantes, como a navegação de forma

simples e rápida por GPS para os locais dos anúncios.

Quando o utilizador visualiza um anúncio, seja através do menu Ver Anúncios, ou do mapa, tem disponível a opção Navegar Para, que automaticamente abre a aplicação de navegação do Android com as coordenadas geográficas do anúncio. Assim é automaticamente calculado o trajeto para o anúncio e são fornecidas todas as indicações ao utilizador ao longo do percurso.

3.3.6 Respostas aos Anúncios

Quando os utilizadores visualizam um anúncio, têm disponível a opção de responder ao anunciante. Esta opção permite que o utilizador escreva e envie uma mensagem diretamente através da aplicação. A mensagem enviada é reencaminhada para o anunciante automaticamente pelo servidor dos GeoAnúncios, através de email. O email contém a identificação do anúncio ao qual a mensagem diz respeito, assim como o email do utilizador que a enviou.

Esta funcionalidade, além de tornar o contacto mais simples e rápido, permite também manter a privacidade do email do anunciante. Quando a resposta é entregue ao anunciante, termina a participação do serviço GeoAnúncios. Se o anunciante decidir responder à mensagem inicial, tem de o fazer por email, diretamente para o utilizador que enviou a mensagem.

3.3.7 Os Meus Anúncios

O menu Os Meus Anúncios disponibiliza as funcionalidades de criação, de publicação e gestão de anúncios. Ao entrar neste menu, é apresentada ao utilizador a lista dos anúncios que publicou anteriormente, podendo utilizar o filtro para organizar a lista por categorias.

O utilizador pode visualizar e editar os anúncios que tem publicados, criar um novo anúncio, ou visualizar os que estão guardados por publicar.

■ Criar Novo Anúncio

Quando um utilizador escolhe a opção de criar um novo anúncio, é-lhe apresentado o formulário de publicação. Neste formulário, é obrigatória a seleção da categoria do anúncio, a definição do título, da descrição e do local do anúncio.

Na fase inicial foi implementado apenas um método para a definição do local do anúncio, que consistia na aquisição da posição atual do utilizador através do GPS. Esta limitação seria uma vantagem já que conduziria os utilizadores a ligarem o GPS, criando uma oportunidade para a recolha de dados. No entanto, constatou-se que isso condicionaria consideravelmente a publicação dos anúncios, já que o utilizador teria que estar no local do anúncio para completar o formulário de publicação. Por este motivo, flexibilizou-se este aspeto, sendo criados dois métodos adicionais para a definição do local, através da seleção de um local no mapa, ou pela

introdução manual da latitude e longitude do local. Após definido o local através de um dos métodos citados, a aplicação utiliza as coordenadas para obter automaticamente, através da Internet, a morada completa do local, que é apresentada no formulário e que faz parte do anúncio.

Além das informações obrigatórias, o utilizador pode adicionar uma imagem da galeria do seu *smartphone* ao anúncio. A imagem cria um efeito visual nos anúncios, sendo um fator diferenciador e pode adicionalmente transmitir outro tipo de informações.

É possível definir uma data limite de disponibilidade para os anúncios. Considerando o caso dos eventos culturais, como por exemplo os concertos, esta funcionalidade é de extrema importância e utilidade. Assim, após a data do concerto, o anúncio deixará de estar disponível automaticamente, sem que tenha de ser manualmente eliminado. O servidor é que determina através da data limite dos anúncios e da data atual, os que já terminaram, tornando-os indisponíveis para os utilizadores.

Se, por algum motivo, o utilizador tiver de interromper a criação de um anúncio, ou não poder publicar o anúncio naquele momento, poderá guarda-los e continuar mais tarde. Quando completo, o utilizador poderá efetuar a publicação, sendo nesta altura necessária a ligação à Internet. Precede ao envio do pedido de publicação, a verificação de todos os campos e opções do anúncio. Se tudo estiver correto e o utilizador possuir créditos suficientes, é iniciado o processo de publicação. Quando selecionada uma imagem, esta é convertida para o formato JPEG e redimensionada para um tamanho mais pequeno, de modo a evitar que o *upload* seja demorado e que consuma grande quantidade de tráfego, mas também para que a posterior visualização dos anúncios seja mais fluida.

Ao receber o pedido de publicação, o servidor insere o novo anúncio na base de dados, desconta os créditos da conta do utilizador, e armazena a imagem no sistema de ficheiros, renomeando-a para o ID único do anúncio, sendo assim feita a associação. O processo termina com o envio da resposta do servidor para a aplicação, confirmando o sucesso da publicação.

■ Edição de Anúncios

O utilizador pode eliminar ou editar as informações dos seus anúncios publicados anteriormente. O formulário de edição de um anúncio é semelhante ao da publicação, sendo necessária a ligação à Internet para editar e atualizar um anúncio. A eliminação torna um anúncio indisponível para consulta. No entanto, não é eliminado completamente da base de dados do servidor, apenas marcado como indisponível, assim mantêm-se todos os anúncios por questões de segurança e de não repúdio.

■ Anúncios Guardados

Consiste numa lista dos anúncios que o utilizador criou, mas ainda não publicou. Estes anúncios podem ser visualizados, editados e quando completos, publicados. Apenas a publicação não se encontra disponível em modo *offline*.

3.3.8 Favoritos

Ao visualizar um anúncio, o utilizador pode adicioná-lo aos seus favoritos. Os favoritos são guardados na base de dados local. Deste modo, o utilizador pode em qualquer altura, mesmo que sem ligação à Internet, visualizar o anúncio completo e utilizar as funções de navegação para o local. Quando desejar, pode remover anúncios marcados como favoritos da lista. Também aqui se encontra disponível o filtro por categoria, de modo a facilitar a organização.

Capítulo 7

Conclusões e Trabalho Futuro

1 Conclusões

O processo de investigação e o trabalho desenvolvido no âmbito da presente dissertação, culmina num interessante conjunto de ilações. A abrangência desta dissertação permitiu a identificação de desafios e de problemas, inerentes aos sistemas de *sensing* que se centram na utilização de dispositivos móveis pessoais.

Desenvolver uma solução de *sensing* deste âmbito, implica abordar áreas distintas, englobando desafios técnico, tecnológicos, mas também sociais.

Este trabalho deu origem a um sistema de *mobile sensing*, que integra um conjunto de soluções que permitem ultrapassar ou minimizar os problemas e desafios que inicialmente foram propostos como objetivos desta dissertação, assim como desafios que surgiram ao longo do processo de investigação e desenvolvimento.

1.1 Processo de Recolha de Dados

O objetivo primário da aplicação desenvolvida consiste na recolha de dados utilizando *smartphones*, a sua centralização e armazenamento num servidor.

■ Definição de uma Arquitetura para o Sistema de Recolha de Dados

Foi especificada e implementada uma arquitetura para o sistema de recolha de dados, que permite dar resposta a um conjunto de requisitos. Nomeadamente:

- Definição dos elementos aos quais os *smartphones* têm acesso (Pontos de acesso, dispositivos Bluetooth, estações base, satélites GPS) e respetivos interfaces que permitem o seu acesso.

- Armazenamento temporário dos dados recolhidos no dispositivo, de modo a endereçar o problema de falta de conectividade para o seu envio.
- Comunicação com o servidor para o envio dos dados e receção de configurações.
- Armazenamento dos dados no servidor.

■ Tecnologias Utilizadas

Foram identificadas as tecnologias disponíveis nos *smartphones* que podem ser utilizadas para recolher dados, assim como quais os dados que é possível recolher utilizando cada uma das tecnologias. Deste modo, foram implementados métodos para a recolha de dados através de Wi-Fi, de Bluetooth, da Rede Móvel Celular e do GPS.

1.2 Poupança de Energia

Como discutido ao longo deste documento, este tipo de aplicações de recolha de dados apresenta um forte impacto na autonomia energética dos dispositivos, dificultando a sua utilização por parte dos utilizadores.

■ Formato dos Dados Enviados e Recebidos

Através da utilização de estruturas JSON, é possível minimizar o *overhead* na troca de mensagens entre a aplicação e o servidor, mas principalmente no envio dos dados recolhidos. Esta opção permite reduzir os consumos energéticos resultantes da utilização dos interfaces de ligação à Internet. Adicionalmente, sendo o *parsing* dos dados JSON mais simples, o seu processamento utiliza menos o CPU, reduzindo o consumo de energia.

■ Amostragem Periódica

Apesar de limitadas a um conjunto específico de parâmetros, como a capacidade da bateria e consumos dos interfaces, que diferem entre dispositivos, as estimativas do consumo de energia tornaram possível concluir que uma abordagem baseada em amostragem contínua seria impraticável. Adicionalmente, permitiram estimar a diferença nos consumos considerando os mesmos parâmetros, de uma amostragem periódica. Este tipo de amostragem permite reduzir consideravelmente o consumo de energia, sendo que o consumo varia de acordo com os períodos de amostragem definidos para cada interface. Os períodos devem ser definidos tendo em conta diversos critérios, como o consumo de energia de cada interface, o seu alcance, entre outros. Deste modo, é importante que os períodos de amostragem possam ser configurados remotamente.

■ Detecção de Movimento

O desenvolvimento do algoritmo de deteção de movimento, que utiliza os dados do acelerómetro, tem como objetivo a obtenção do equilíbrio entre a eficácia da recolha de dados e o consumo energético. Este é, sem dúvida, um dos maiores desafios e um dos problemas críticos das aplicações de *mobile sensing*. Se, por um lado, se pretende manter a eficácia da recolha de dados, detetando todas as situações em que o utilizador está em movimento, a eficiência energética é por consequência afetada, e vice-versa.

O algoritmo desenvolvido elimina os consumos nos períodos de imobilidade do dispositivo, representando uma significativa poupança de energia. No entanto, os consumos diários da aplicação estão dependentes como referido dos consumos que diferem entre os dispositivos, mas também do tempo que o utilizador está em movimento.

■ Gestão da Atividade do CPU

É sem dúvida necessário gerir a utilização do CPU. O sistema Android, assim como outros sistemas operativos móveis, adormecem o CPU quando o telefone está inativo, o que é um grande problema para este tipo de aplicações. Se a utilização de mecanismos como o *WakeLock*, que permite manter o CPU ativo nestas situações, resolve de certa forma o problema, a consequência é um inaceitável consumo de energia. Deste modo, o *WakeLock* não pode ser utilizado na sua forma básica, é necessário gerir a sua utilização. Com esse objetivo, foi implementado um mecanismo adicional. Através de um alarme do sistema, que é difundido mesmo quando o telefone está em modo inativo, é possível acordar o CPU periodicamente, sendo feita a análise de movimento. Se existir movimento é feita a recolha dos dados, caso contrário o CPU volta a adormecer. Desta forma é possível reduzir os consumos elevados da atividade constante do CPU, com um impacto residual na eficácia da recolha de dados.

1.3 Segurança, Privacidade e Integridade

Na maioria das aplicações de *sensing*, a proteção dos dados não é considerada, por consequência a privacidade dos utilizadores é posta em causa. Esta questão é muito importante não só para os utilizadores, mas também para a instituição responsável pelo sistema de *sensing*. Se um sistema de *sensing* apresentar garantias da proteção dos dados, a adesão dos utilizadores será superior, resultando no aumento da massa crítica. Por outro lado, a integridade dos dados é importante para a coerência e veracidade dos estudos, pois se os dados estiverem vulneráveis podem adulterados ou fabricados por agentes externos ao sistema.

Foram tomadas decisões e implementados mecanismos que direta e indiretamente se apresentam como resposta a estes problemas.

■ Forma de Identificação dos Dados

É necessário identificar a fonte dos dados univocamente, para que seja possível a diferenciação dos padrões de movimento. A identificação não deve ser baseada em informação pessoal. No caso dos *smartphones*, a identificação pode ser conseguida através dos endereços MAC dos interfaces de Bluetooth e Wi-Fi, ou através do ID único do dispositivo.

■ Privacidade dos Dados de Registo

As aplicação que requerem registo, como é o caso dos GeoAnúncios, devem salvaguardar a privacidade dos dados dos utilizadores. Assim, a password introduzida pelo utilizador não é armazenada em texto limpo, não sendo possível aos administradores do sistema, ou a um intruso, o conhecimento da mesma. A maioria dos serviços que requerem registo não mencionam quais os mecanismos que utilizam para proteger os dados.

Além de salvaguardar o sistema em caso de ataque, referenciar aos utilizadores a utilização destes mecanismos pode fazer diferença na confiança no sistema.

■ Separação das Bases de Dados

A separação das bases de dados, do serviço prestado aos utilizadores e dos dados recolhidos, é importante por diversos fatores.

Em primeiro lugar, o acesso a uma das bases de dados não implica o acesso à outra. No caso da análise dos dados recolhidos ser feita no âmbito de outros estudos ou instituições, pode ser dado o acesso apenas aos dados recolhidos, salvaguardando os conteúdos do serviço, como por exemplo os emails dos utilizadores.

Em segundo lugar, torna os sistemas independentes, podendo ser distribuídas outras aplicações que fornecem serviços distintos, mas que utilizam o mesmo módulo de recolha de dados e a mesma base de dados de armazenamento.

■ Utilização de Métodos Criptográficos

A utilização de métodos criptográficos, para a cifragem dos dados enviados para o servidor, é essencial. A utilização de um algoritmo como o AES-128 bits, como o que foi utilizado neste projeto, é suficiente para garantir um nível de proteção dos dados adequado. Sendo um algoritmo de chave simétrica, a sua segurança baseia-se no tamanho da chave e em manter a mesma secreta.

Poderia ser utilizada uma chave de 256 bits, indicada para informações altamente secretas. No entanto, introduziria mais *overhead* e aumentaria o tempo de processamento. Considerando os consumos energéticos nos dispositivos móveis e a sensibilidade dos dados, a utilização de 128 bits é uma opção coerente.

1.4 Motivação dos Utilizadores

O desafio de motivar as pessoas a utilizar uma aplicação de *sensing* foi intensamente discutido ao longo desta dissertação. Idealiza-se que a solução para este problema não pode passar por ocultar a recolha de dados dos utilizadores, mas por assumir uma posição séria, criando formas de recompensa motivantes e ganhando a confiança das pessoas, estabelecendo assim uma relação de mútuo interesse.

■ Modelo de Recompensa

O modelo de recompensa é sem dúvida um ponto chave para definir o tipo de aplicação a desenvolver. Concluiu-se, que um modelo baseado num serviço como recompensa, é o mais indicado para aplicações de *sensing*.

A atribuição de créditos aos utilizadores, que surge como forma de recompensa adicional pela recolha de dados, permite não só justificar em certa medida a recolha de dados, mas também conferir aos utilizadores uma sensação de posse de um recurso que podem utilizar, mesmo que virtualmente.

■ Funcionalidades

A aplicação desenvolvida engloba várias funcionalidades, que a tornam uma ferramenta útil para a publicação e procura de anúncios. A sua abrangência de conteúdos engloba diversos

interesses, maximizando o público alvo. Permite a procura emprego, a divulgação de negócios e estabelecimentos, mas também de eventos e locais lúdicos, como concertos, diversão noturna ou um bom restaurante.

A georreferenciação dos anúncios é uma característica chave desta aplicação, sendo uma das vantagens em relação aos restantes serviços de anúncios, como por exemplo os web sites. Esta característica, torna possível disponibilizar funcionalidades que outras aplicações não oferecem, como a visualização dos anúncios pelo mapa, a possibilidade dos utilizadores se deslocarem por uma cidade descobrindo anúncios perto de si, assim como a possibilidade de navegar para o local do anúncio utilizando o GPS.

As aplicações devem permitir o acesso a um serviço útil e interessante, com um ponto chave que o distinga dos restantes serviços disponíveis.

■ Impacto Visual

O interface gráfico foi um ponto da aplicação que recebeu considerável atenção. Como referido anteriormente, muitos projetos apenas focam os seus esforços no desenvolvimento das funcionalidades da aplicação, desprezando a importância do interface.

Assume-se uma posição clara neste aspeto, quando uma aplicação pretende atrair utilizadores, o interface têm de ser atrativo e intuitivo. O interface é a "cara" da aplicação, devendo despertar de imediato o interesse por parte dos utilizadores.

■ Disponibilização e Divulgação

A divulgação da aplicação é essencial para que a aplicação dê os primeiros passos e para o seu crescimento futuro. Foi desenvolvido um web site oficial e criados cartazes digitais para enviar por email. O web site e os emails de divulgação são a primeira interação com os possíveis utilizadores, devendo ser apelativos.

Numa primeira fase, a aplicação foi disponibilizada apenas através do site oficial, sendo realizada a primeira divulgação, pela comunidade académica, no dia 27 de Setembro de 2012. Posteriormente, no dia 8 de Outubro de 2012, a aplicação foi disponibilizada no Google Play, sendo realizada nova divulgação pela comunidade académica realçando esta novidade.

2 Análise dos Resultados

O período para a recolha de resultados foi curto, devido à limitação temporal imposta pela data de entrega da dissertação. No entanto, os resultados que serão apresentados de seguida, são suficientes para analisar a adesão à aplicação e os dados que foi possível recolher durante esse período de tempo.

2.1 Adesão à Aplicação

No gráfico da figura 7.1, é apresentado o número de downloads, registos diários e total de utilizadores do serviço GeoAnúncios.

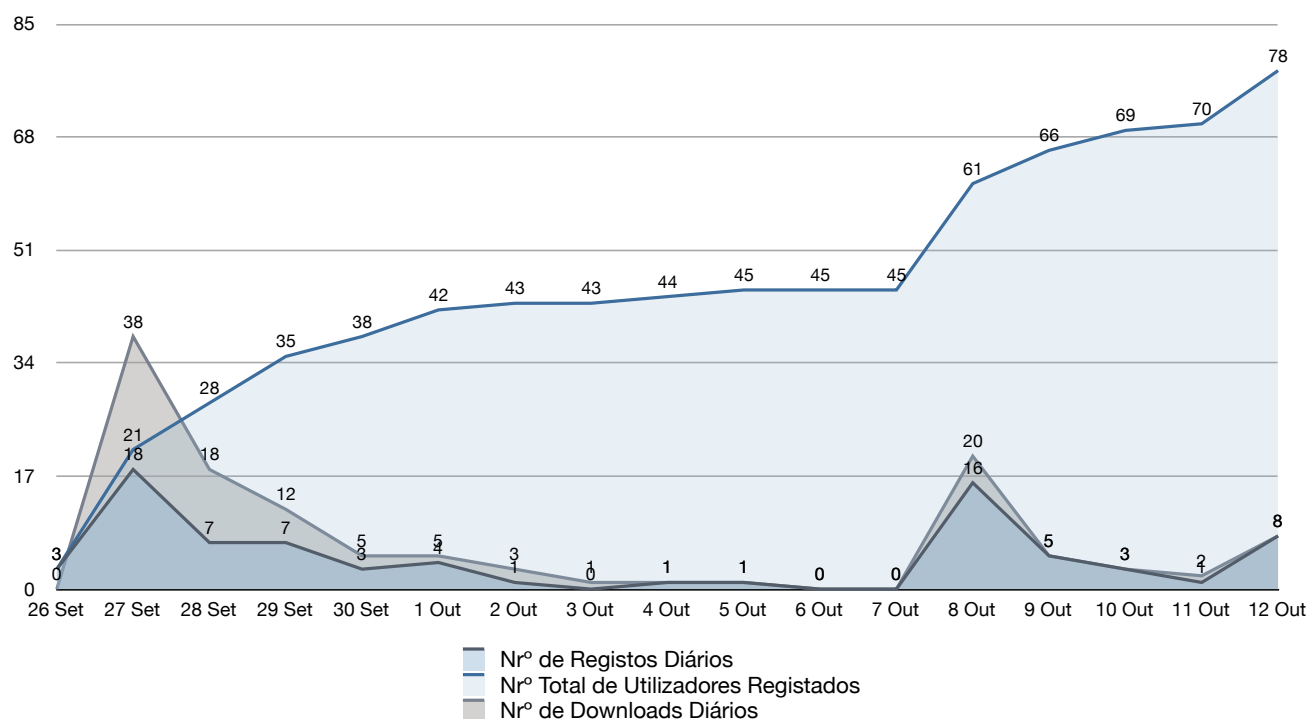


Figura 7.1: Estatísticas de Downloads, Registos e Total de Utilizadores

Após a primeira divulgação, que se realizou no dia 27 de Setembro de 2012, verificou-se que nos quatro dias seguintes, a aplicação foi descarregada 73 vezes, verificando-se 42 utilizadores registados. O principal motivo para o desfasamento entre o número de downloads e o número de registos, deveu-se eventualmente a uma configuração padrão dos dispositivos Android. Na maioria dos dispositivos, as configurações de origem permitem apenas a instalação de aplicações através do Google Play. O contacto de um utilizador que não conseguia instalar a aplicação por este motivo, alertou-nos para este facto. Muitos utilizadores não conseguem ou não querem, por receios de segurança, alterar esta definição.

Verificou-se igualmente que alguns utilizadores instalavam a aplicação mas não se registavam. Este comportamento, poderia dever-se à falta de confiança na aplicação, por não estar disponível no Google Play, ou devido aos termos de utilização que fazem referência à recolha de dados. Para resolver tanto a dificuldade dos utilizadores em instalarem a aplicação, como para que os utilizadores tivessem mais confiança na aplicação, optou-se por disponibilizá-la no Google Play. A aplicação não foi inicialmente disponibilizada no Google Play porque ainda não tinha sido testada a sua estabilidade em situações de elevada utilização.

Posteriormente à disponibilização no Google Play, foi realizada uma nova divulgação pela comunidade académica no dia 08 de Outubro de 2012. A aplicação foi descarregada mais 38 vezes até ao dia 12 de Outubro 2012, registando-se 33 novos utilizadores. Verificou-se desta forma, que o desfasamento entre o número de downloads e de registos na primeira divulgação, se deveu em grande parte à aplicação não estar no Google Play. Entre o site e o Google Play, até ao dia 12 a aplicação foi descarregada 122 vezes num período de 17 dias, com um total de 78 utilizadores registados, sendo que 67 estão ativos regularmente no sistema. Como era esperado, o maior número de registo e downloads ocorreram nos dias em que se realizaram as divulgações, o que evidencia a importância da divulgação de novos serviços e aplicações.

Dado o curto espaço de tempo disponível, para a disponibilização e divulgação da aplicação, o número de utilizadores registados é bastante satisfatório. No entanto, foi limitado por alguns fatores.

A divulgação focou-se apenas na comunidade académica. Uma divulgação mais abrangente permitirá pensarmos nós, que a aplicação comece a ser conhecida e a ser utilizada pela população em geral. O âmbito da aplicação, ou seja a publicação de anúncios e consulta, não é principalmente direcionada para a comunidade académica.

Como foi apresentado anteriormente (Fig. 3.4), apesar da plataforma Android ser a mais utilizada, existe uma elevada utilização de outros dispositivos. A população divide-se entre utilizadores de dispositivos Android, iOS, Symbian, entre outros. Por este motivo, apesar da divulgação atingir um elevado número de pessoas na comunidade académica, foi apenas direcionada aos utilizadores de Android, o que limita em grande escala os possíveis utilizadores da aplicação.

O registo e os termos de utilização, revelam sem dúvida impacto no registo de utilizadores. Após a disponibilização no Google Play, o desfasamento entre o número de downloads e o número de registos diminuiu drasticamente, no entanto continuou a existir, esta situação comprova que o registo influencia a adesão das pessoas. Diversas aplicações disponíveis mesmo no Google Play, recolhem dados ocultando essa informação. Como referido, o objetivo deste

projeto é encarar este problema de frente, sendo transparente com os utilizadores. Só assim será possível tornar as aplicações de *sensing* confiáveis dissipando os receios dos utilizadores, o que constitui um dos maiores problemas.

As aplicações carecem de um período de amadurecimento e crescimento, principalmente as que se baseiam em serviços. Serviços como o Facebook, que são utilizados massivamente, percorreram um longo caminho até chegarem ao patamar em que hoje se encontram.

Estas limitações podem e devem ser estudadas, de modo a encontrar soluções e a atingir resultados ainda mais satisfatórios no número de utilizadores.

2.2 Dados Recolhidos

O gráfico da figura 7.2, representa o número de amostras recolhidas e armazenadas no servidor, segundo cada interface. Na tabela 7.1, são apresentados os períodos de amostragem para cada interface.

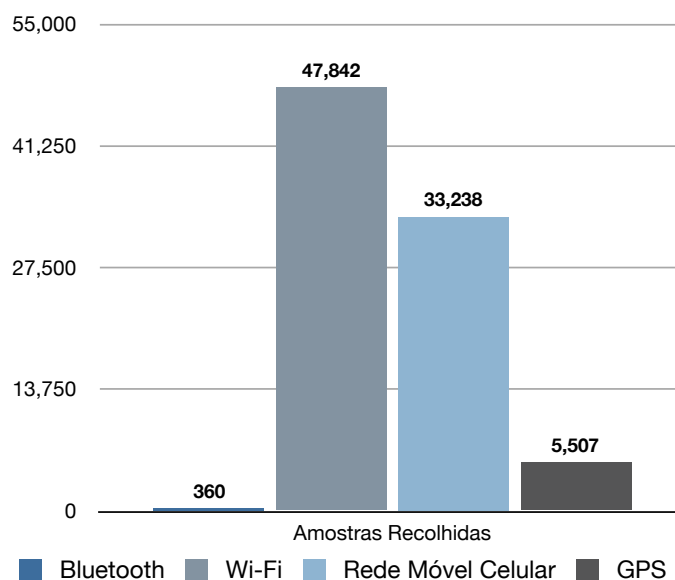


Figura 7.2: Amostras recolhidas de cada Interface

Interface	Período de Amostragem
Wi-Fi	10 segundos
GSM	20 segundos
Bluetooth	20 segundos
GPS	60 segundos

Tabela 7.1: Períodos utilizados na recolha de dados

Os resultados apresentados (Fig. 7.2) permitem concluir que a disponibilidade do interface Bluetooth para recolha de dados é muito reduzida. Por ser um interface pouco utilizado, normalmente é mantido desligado pelos utilizadores dos dispositivos.

Apesar do período de amostragem do GPS ser três vezes superior, o número de amostras recolhidas é significativamente superior às de Bluetooth, o que indica que o GPS está disponível para recolha consideravelmente mais tempo que o Bluetooth. Em relação ao Wi-Fi e ao GSM, apesar do período de amostragem do Wi-Fi ser metade do período de amostragem do GSM, a diferença no número de amostras recolhidas não é o dobro. Estes resultados são compreensíveis, visto que alguns utilizadores desligam o Wi-Fi quando não o estão a utilizar para poupar energia, sendo que o GSM está constantemente disponível.

Verificou-se que alguns utilizadores apesar de utilizarem o serviço regularmente, não obtiveram créditos, ou obtiveram poucos quando comparados com a média. Isto significa que em alguns casos, os dados recolhidos não são enviados para o sistema, ou são poucas vezes enviados. Esta situação ocorre porque, como referido, optou-se por não enviar os dados através da Rede Móvel Celular de modo a não consumir o tráfego dos utilizadores, que normalmente é limitado. Alguns utilizadores acedem ao serviço GeoAnúncios pela Rede Móvel Celular, permanecendo ativos no sistema, no entanto os dados recolhidos não são enviados. Se todos os utilizadores enviassem os dados recolhidos regularmente, o número de amostras armazenadas no servidor seria ainda mais elevado. O número de amostras varia muito rapidamente, basta que um utilizador, que não teve conectividade com o servidor durante alguns dias, envie os dados armazenados no seu dispositivo.

Estas conclusões e resultados, podem auxiliar outros projetos da área, que certamente se deparam com os mesmos desafios e problemas. Foi de extrema importância testar a aplicação em contexto real, sendo possível retirar ilações interessantes sobre o comportamento dos utilizadores em relação às aplicações de *mobile sensing*.

3 Trabalho Futuro

Encontra-se em preparação um artigo a ser publicado brevemente, que expõe as ilações concernentes às matérias mais pertinentes abordadas no âmbito desta dissertação.

Alguns dos desafios abordados ao longo desta dissertação são complexos, sendo essencial um estudo mais aprofundado a fim de otimizar as soluções apresentadas. Este trabalho pode evoluir em diversos aspetos, sendo que algumas ideias ficam por implementar devido à limitação temporal inata ao processo de dissertação.

Em relação à recolha de dados, é importante que seja realizado um estudo mais exaustivo dos períodos de amostragem para cada interface, dada a sua influência na correlação entre o consumo energético e a eficácia da recolha de dados.

O algoritmo de deteção de movimento pode evoluir de modo a detetar o movimento do utilizador com mais precisão, por exemplo recorrendo à análise do ambiente em redor do utilizador para ajustar dinamicamente os valores fronteira. Adicionalmente, quando o nível de bateria do *smartphone* atingir um determinado valor (por exemplo 20%) a recolha de dados pode ser suspensa, evitando que a bateria seja totalmente consumida pela recolha de dados. Pode ainda ser realizada uma análise considerando os dados recolhidos dos diferentes interfaces e a percentagem de bateria consumida para um determinado intervalo de tempo. Assim, se o consumo de bateria for elevado e não tiverem sido recolhidos dados significativos de certos interfaces, a recolha de dados através destes pode ser suspensa temporariamente, ou alargado o seu período de amostragem, reduzindo desta forma o consumo regular de energia de um interface que não está a fornecer dados significativos.

Os dados armazenados no servidor podem ser mantidos encriptados até que sejam analisados, deste modo são mantidos protegidos, salvaguardando o sistema de ataques.

Relativamente à aplicação e serviço GeoAnúncios, como especificado na arquitetura do sistema, seria interessante disponibilizar algumas das funcionalidades através de um web site, de modo dinamizar o serviço envolvendo os utilizadores.

O registo, poderia ser um requisito apenas para a publicação de anúncios ou para o envio de uma resposta, minimizando assim o seu efeito psicológico nos utilizadores.

Além do modo *offline*, a base de dados local poderia ser utilizada como *cache* para todo o sistema, aumentando a fluidez na visualização dos anúncios, principalmente no mapa, que é um processo mais demorado que aumenta com o elevado número de anúncios disponíveis.

Poderiam ser adicionados informações adicionais e conteúdos dinâmicos aos anúncios, como

vídeos promocionais de concertos. Adicionalmente, a resposta a anúncios poderia ser melhorada, sendo integrada na aplicação um sistema completo de troca de mensagens.

Finalmente, sendo um serviço que está dependente dos conteúdos disponíveis e considerando que se verificam mais utilizadores a consultar do que a publicar anúncios, seria importante implementar um sistema que automaticamente introduzisse conteúdos disponíveis na Internet, ou através de uma parceria com um serviço de anúncios, de modo a garantir o fluxo de novos conteúdos.

Bibliografia

- [AAB⁺07] Tarek Abdelzaher, Yaw Anokwa, Péter Boda, Jeff Burke, Deborah Estrin, Leonidas Guibas, Aman Kansal, Sam Madden, and Jim Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, 6:20–29, 2007.
- [adi12] Webservices, J2EE and moreT. http://adityakeyal.blogspot.pt/2011/03/what-are-rest-soap-json_04.html/, 2012. Última consulta a 15 de Agosto de 2012.
- [ali10] Android open source project: Philosophy and goals. <http://source.android.com/about/philosophy.html>, 2010. Última consulta a 15 de Agosto de 2012.
- [ali12] Open handset alliance: Android overview. http://www.openhandsetalliance.com/android_overview.html, 2012. Última consulta a 15 de Agosto de 2012.
- [And10] Andy. Apple Bans Wi-Fi Stumbler iPhone Apps – Blames Use Of Private APIs. <http://www.iphonehacks.com/2010/03/apple-bans-wifi-stumbler-iphone-apps-blames-use-of-private-apis.html>, Março 2010. Última consulta a 15 de Agosto de 2012.
- [And12a] Android. Android Developer Reference. <http://developer.android.com/reference/packages.html>, 2012. Última consulta a 15 de Agosto de 2012.
- [And12b] Android. Application Fundamentals. <http://developer.android.com/guide/components/fundamentals.html>, 2012. Última consulta a 30 de Agosto de 2012.
- [App08] Apple. Revolutionary phone. <http://web.archive.org/web/20080111051348/http://www.apple.com/iphone/features/index.html>, Janeiro 2008. Última consulta a 15 de Agosto de 2012.

- [App12a] Apple. App store approval guidelines. <https://developer.apple.com/appstore/resources/approval/guidelines.html>, 2012. Última consulta a 15 de Agosto de 2012.
- [App12b] Apple. App store review guidelines. <https://developer.apple.com/appstore/guidelines.html>, 2012. Última consulta a 15 de Agosto de 2012.
- [App12c] Apple. Apple: ios developer program - distribute. 2012. Última consulta a 15 de Agosto de 2012.
- [App12d] Apple. Apple keynote: 11 junho de 2012. <http://www.apple.com/apple-events/june-2012/>, Junho 2012. Última consulta a 15 de Agosto de 2012.
- [App12e] Apple. ios developer program - develop. <https://developer.apple.com/programs/ios/develop.html>, 2012. Última consulta a 15 de Agosto de 2012.
- [App12f] Apple. iPhone 5 Features. <http://www.apple.com/iphone/features/>, Setembro 2012. Última consulta a 12 de Setembro de 2012.
- [ARM12] ARM. Getting Started with Android. <http://www.arm.com/community/software-enablement/google/solution-center-android/getting-started-with-android-on-arm.php>, 2012. Última consulta a 30 de Agosto de 2012.
- [BHN04] T.D. Bui, D. Heylen, and A. Nijholt. Combination of facial movements on a 3d talking head. In *Computer Graphics International, 2004. Proceedings*, pages 284 –290, june 2004.
- [Bla12] Rafe Blandford. 100,000 apps published to Windows Phone Marketplace. http://allaboutwindowsphone.com/news/item/14960_100000_apps_published_to_Windo.php, Junho 2012. Última consulta a 15 de Agosto de 2012.
- [Blu12] Bluetooth.com. Bluetooth.com. <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>, 2012. Última consulta a 10 de Abril de 2012.

- [BT08] Dirk Brockmann and Fabian J. Theis. Money circulation, trackable items, and the emergence of universal human mobility patterns. *IEEE Pervasive Computing*, 7(4):28–35, 2008.
- [Can11] Canalys. Google’s android becomes the world’s leading smart phone platform. <http://www.canalys.com/newsroom/google%E2%80%99s-android-becomes-world%E2%80%99s-leading-smart-phone-platform>, Janeiro 2011. Última consulta a 15 de Agosto de 2012.
- [CC11] J. Chon and Hojung Cha. Lifemap: A smartphone-based context provider for location-based services. *Pervasive Computing, IEEE*, 10(2):58–67, feb. 2011.
- [CCR10] I. Constandache, R.R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, march 2010.
- [CEL⁺06] Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, and Ronald A. Peterson. People-centric urban sensing. In *In The Second Annual International Wireless Internet Conference (WICON*, pages 2–5. IEEE Computer Society Press, 2006.
- [CEN12] CENS. Cens urban sensing project. http://research.cens.ucla.edu/projects/2006/Systems/Urban_Sensing/default.htm, 2012. Última consulta a 11 de Abril de 2012.
- [com12] comScore. Press Release - 2012 Mobile Future in Focus. http://www.comscore.com/Press_Events/Press_Releases/2012/2/comScore_Releases_the_2012_Mobile_Future_in_Focus_Report?piCId=66038, Maio 2012. Última consulta a 08 de Agosto de 2012.
- [Cov12] Cover Pages. XML Applications and Initiatives. <http://xml.coverpages.org/xmlApplications.html>, 2012. Última consulta a 15 de Agosto de 2012.
- [CSM⁺05] Sunny Consolvo, Ian E. Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location disclosure to social relations: why,

- when, & what people want to share. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 81–90, New York, NY, USA, 2005. ACM.
- [CTSC11] Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 82–95, New York, NY, USA, 2011. ACM.
- [DA00] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Workshop on The What, Who, Where, When, and How of Context-Awareness (CHI 2000)*, The Hague, The Netherlands, April 2000.
- [Dai11] DailyChanges. Programming Language Popularity. <http://langpop.com/>, Abril 2011. Última consulta a 30 de Agosto de 2012.
- [Dai12] DailyChanges. DailyChanges. <http://www.dailychanges.com/>, Agosto 2012. Última consulta a 30 de Agosto de 2012.
- [Dem10] M. Demirbas. Invited talk: Employing the smartphone (and the smart human) as the mobile sensor. In *Distributed Computing in Sensor Systems Workshops (DCOSSW), 2010 6th IEEE International Conference on*, page 1, june 2010.
- [DK01] Qizhi Dai and R.J. Kauffman. Business models for internet-based e-procurement systems and b2b electronic markets: an exploratory assessment. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 10 pp., jan. 2001.
- [EML⁺07] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G s. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *In Proc. of 5th ACM Conf. on Embedded Networked Sensor Systems*, pages 6–9, 2007.
- [ER06] Ahmed El-Rabbany. *Introduction to GPS: The Global Positioning System, Second Edition*. Artech House Publishers, August 2006.

- [erg12] Ergo: On-the-go air quality readings delivered to your mobile device. <http://www.urban-atmospheres.net/Experiments/Ergo/index.html>, 2012. Última consulta a 10 de Abril de 2012.
- [Fox05] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, 25(6):38–46, nov.-dec. 2005.
- [Fra12] Steve Francia. spf13: SOAP vs REST. <http://spf13.com/post/soap-vs-rest/>, 2012. Última consulta a 15 de Agosto de 2012.
- [Gar12] Gartner: Technology Research and Business Leader Insight. <http://www.gartner.com/it/page.jsp?id=2017015>, Maio 2012. Última consulta a 08 de Agosto de 2012.
- [GHB08] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.
- [GKN⁺07] W.I. Grosky, A. Kansal, S. Nath, Jie Liu, and Feng Zhao. Senseweb: An infrastructure for shared sensing. *Multimedia, IEEE*, 14(4):8–13, oct.-dec. 2007.
- [GKT10] Björn Gressmann, Helge Sören Klimek, and Volker Turau. Towards ubiquitous indoor location based services and indoor navigation. In *WPNC*, pages 107–112. IEEE, 2010.
- [GM07] A. Gupta and P. Mohapatra. Energy consumption and conservation in wifi based phones: A measurement-based study. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, pages 122–131, june 2007.
- [Gol04] Robin F. Goldsmith. *Discovering real business requirements for software project success*. Artech House, 2004.
- [Gol12] Boris Golden. What is Systems Architecture? http://www.lix.polytechnique.fr/~golden/systems_architecture.html#principles, 2012. Última consulta a 2 de Agosto de 2012.

- [Haa12] Juergen Haas. Modular programming. <http://linux.about.com/cs/linux101/g/modularprogramm.htm>, 2012. Última consulta a 15 de Agosto de 2012.
- [HBZ⁺06] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In *In 4th ACM SenSys*, pages 125–138, 2006.
- [HST10] A. Hammershoj, A. Sapuppo, and R. Tadayoni. Challenges for mobile application development. In *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference on*, pages 1–8, oct. 2010.
- [HT07] Harri Holma and Antti Toskala. *WCDMA for UMTS: HSPA Evolution and LTE*. John Wiley & Sons, Inc., New York, NY, USA, 2007.
- [IDC11] IDC: International Data Corporation. Press Release: Worldwide Smartphone Market Expected to Grow 552011 and Approach Shipments of One Billion in 2015. <http://www.idc.com/getdoc.jsp?containerId=prUS22871611>, Junho 2011. Última consulta a 08 de Agosto de 2012.
- [IDC12a] IDC: International Data Corporation. Press Release: Android- and iOS-Powered Smartphones Expand Their Share of the Market in the First Quarter. <http://www.idc.com/getdoc.jsp?containerId=prUS23503312>, Maio 2012. Última consulta a 08 de Agosto de 2012.
- [IDC12b] IDC: International Data Corporation. Press Release: Android Expected to Reach Its Peak This Year as Mobile Phone Shipments Slow. <http://www.idc.com/getdoc.jsp?containerId=prUS23523812>, Junho 2012. Última consulta a 08 de Agosto de 2012.
- [IDC12c] IDC: International Data Corporation. Press Release: Apple Shines, But Slumping Android Shipments Lead to Disappointing First Quarter for Media Tablets, According to IDC . <http://www.idc.com/getdoc.jsp?containerId=prUS23466712>, Maio 2012. Última consulta a 15 de Agosto de 2012.
- [IDC12d] IDC: International Data Corporation. Press Release: Western European Mobile Phone Market Returns to Growth in 1Q12. <http://www.idc.com/>

getdoc.jsp?containerId=prUK23507512, Maio 2012. Última consulta a 08 de Agosto de 2012.

- [IDC12e] IDC: International Data Corporation. Press Release: Worldwide Smartphone Market Continues to Soar. <http://www.idc.com/getdoc.jsp?containerId=prUS23455612>, Maio 2012. Última consulta a 08 de Agosto de 2012.
- [IMA12] Inside Mobile Apps: iOS leapfrogs Android with 410 million devices sold and 650,000 apps. <http://www.insidemobileapps.com/2012/07/24/ios-device-sales-leapfrog-android-with-410-million-devices-sold/> 2012. Última consulta a 15 de Agosto de 2012.
- [IMS11] IMS Research. Press Release: Global Smartphones Sales Will Top 420 Million Devices in 2011. http://imsresearch.com/press-release/Global_Smartphones_Sales_Will_Top_420_Million_Devices_in_2011_Taking_28_Percent_of_all_Handsets_According_to_IMS_Research, Julho 2011. Última consulta a 08 de Agosto de 2012.
- [ios12] iOS Developer Library: iOS Technology Overview. http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898-CH1-SW1, 2012. Última consulta a 15 de Agosto de 2012.
- [JBG⁺09] Klas Johansson, Johan Bergman, Dirk Gerstenberger, Mats Blomgren, and Anders Wallén. Multi-carrier hspa evolution. In *VTC Spring*, 2009.
- [Jos11] Anthony D. Joseph. Large-scale opportunistic sensing. *Pervasive Computing, IEEE*, 10(4):54–58, april 2011.
- [Kak03] T. Kakola. Software business models and contexts for software innovation: key areas software business research. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, page 8 pp., jan. 2003.
- [KBVM09] Damian Kelly, Ross Behan, Rudi Villing, and Seán McLoone. Computationally tractable location estimation on wifi enabled mobile phones. *Proce-*

edings of the IET Irish Signals and Systems Conference, pages 1–6, June 2009.

- [KCRB09] Gautier Krings, Francesco Calabrese, Carlo Ratti, and Vincent D. Blondel. Scaling behaviors in the communication network between cities. In *CSE (4)*, pages 936–939. IEEE Computer Society, 2009.
- [KKT09] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic sensing: Security challenges for the new paradigm. In *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, pages 1 –10, jan. 2009.
- [KMP12] E. Koukoumidis, M. Martonosi, and Li-Shiuan Peh. Leveraging smartphone cameras for collaborative road advisories. *Mobile Computing, IEEE Transactions on*, 11(5):707 –723, may 2012.
- [KPN96] Richard Knoblauch, Martin Pietrucha, and Marsha Nitzburg. Field studies of pedestrian walking speed and start-up time. *Transportation Research Record*, 1538:27 – 38, 1996.
- [LAI⁺06] Mia Lähteenmäki, Aino Ahtinen, Olli Immonen, Kaj Mäkelä Chand Malu, Petteri J. Saarinen, Mikko Tarkiainen, and Severi Uusitalo. Mobile workforce services opportunity. *Nokia Research Center Helsinki*, November 20 2006.
- [Lan09] Marc Lankhorst, editor. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, 2. edition, 2009.
- [LD10] Sian Lun Lau and K. David. Movement recognition using the accelerometer in smartphones. In *Future Network and Mobile Summit, 2010*, pages 1 –9, june 2010.
- [LEM⁺08] Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T. Campbell. Urban sensing systems: opportunistic or participatory? In *Proceedings of the 9th workshop on Mobile computing systems and applications*, HotMobile ’08, pages 11–16, New York, NY, USA, 2008. ACM.

- [Len09] Philipp Lenssen. Google Announces Definite End of Their SOAP Search API. <http://blogoscoped.com/archive/2009-03-03-n84.html>, 2009. Última consulta a 15 de Agosto de 2012.
- [LH10] Juong-Sik Lee and Baik Hoh. Sell your experiences: a market mechanism based incentive for participatory sensing. In *PerCom*, pages 60–68. IEEE Computer Society, 2010.
- [LKaD⁺10] Sian Lun Lau, I. Kö andnig, K. David, B. Parandian, C. Carius-Dü andssel, and M. Schultz. Supporting patient monitoring using activity recognition with a smartphone. In *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pages 810 –814, sept. 2010.
- [LL09] Yih-Jiun Lee and Kai-Wen Lien. Location based enabled context awareness information service. In *New Trends in Information and Service Science, 2009. NISS '09. International Conference on*, pages 944 –947, 30 2009-july 2 2009.
- [LLC11] Hsien-I Lin, Yu-Cheng Liu, and Chi-Li Chen. Evaluation of human-robot arm movement imitation. In *Control Conference (ASCC), 2011 8th Asian*, pages 287 –292, may 2011.
- [LPL⁺09] Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. Soundsense: Scalable sound sensing for people-centric applications on mobile phones, 2009.
- [LY09] Feida Lin and Weiguo Ye. Operating system battle in the ecosystem of smartphone industry. In *Information Engineering and Electronic Commerce, 2009. IEEEC '09. International Symposium on*, pages 617 –621, may 2009.
- [Men10] Gary Menezes. Symbian os, now fully open source. <http://www.watblog.com/2010/02/06/symbian-os-now-fully-open-source/>, 2010. Última consulta a 15 de Agosto de 2012.
- [Met12] MetroSense. Metrosense. <http://metrosense.cs.dartmouth.edu/>, 2012. Última consulta a 20 de Abril de 2012.

- [MLEC07] Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, and Andrew T. Campbell. Cenceme: Injecting sensing presence into social networking applications. In *in EuroSSC, ser. Lecture Notes in Computer Science*, pages 1–28. Online]. Available: <http://dblp.uni-trier.de/db/conf/eurossc/eurossc2007.html>MiluzzoLEC07, 2007.
- [MM09] Martin Mladenov and Michael Mock. A step counter service for java-enabled devices using a built-in accelerometer. In *Proceedings of the 1st International Workshop on Context-Aware Middleware and Services: affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009)*, CAMS '09, pages 1–5, New York, NY, USA, 2009. ACM.
- [MMRFSPJBM08] Javier Martin-Moreno, Daniel Ruiz-Fernandez, Antonio Soriano-Paya, and Vicente Jesus Berenguer-Miralles. Monitoring 3d movements for the rehabilitation of joints in physiotherapy. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4836 –4839, aug. 2008.
- [Mol11] Brad Molen. Nokia announces Symbian Belle alongside three new devices. <http://www.engadget.com/2011/08/24/nokia-announces-symbian-belle-running-on-three-new-devices/>, Agosto 2011. Última consulta a 15 de Agosto de 2012.
- [MTZ05] S. Madhani, M. Tauil, and Tao Zhang. Collaborative sensing using uncontrolled mobile devices. In *Collaborative Computing: Networking, Applications and Worksharing, 2005 International Conference on*, page 8 pp., 0-0 2005.
- [Muh12] Samiullah Muhammed. Advantages of becoming a PHP Web Developer. <http://www.phphunter.com/2012/07/advantages-of-becoming-php-web.html#.UD9kH2jyYQQ>, 2012. Última consulta a 15 de Agosto de 2012.
- [Mus11] John Musser. Open APIs: State of the Market. <http://www.slideshare.net/jmusser/pw-glue-conmay2010>, 2011. Última consulta a 15 de Agosto de 2012.

- [MW05] P. McDermott-Wells. What is bluetooth? *Potentials, IEEE*, 23(5):33 – 35, 2004-jan. 2005.
- [Nok12] Nokia. Symbian platform: Symbian overview. <http://www.developer.nokia.com/Devices/Symbian/>, 2012. Última consulta a 15 de Agosto de 2012.
- [par10] Participatory urbanism participatory urbanism: Empowering citizens to collectively author, share, and remix measurments from their environment. <http://www.urban-atmospheres.net/ParticipatoryUrbanism/index.html>, 2010. Última consulta a 10 de Abril de 2012.
- [PBB09] Raluca Ada Popa, Hari Balakrishnan, and Andrew Blumberg. VPriv: Protecting Privacy in Location-Based Vehicular Services. In *18th USENIX Security Symposium*, Montreal, Canada, August 2009.
- [PBN11] M.T. Pourazad, A. Bashashati, and P. Nasiopoulos. A random forests-based approach for estimating depth of human body gestures using a single video camera. In *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pages 649 –650, jan. 2011.
- [Pop11] K.M. Popp. Software industry business models. *Software, IEEE*, 28(4):26 –30, july-aug. 2011.
- [Por12] Bluetooth Developer Portal. Architecture - Baseband. <http://developer.bluetooth.org/KnowledgeCenter/TechnologyOverview/Pages/Baseband.aspx>, 2012. Última consulta a 25 de Agosto de 2012.
- [PSW⁺01] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *Wireless Networks*, pages 189–199, 2001.
- [RB07] O. Riva and C. Borcea. The urbanet revolution: Sensor power to the people! *Pervasive Computing, IEEE*, 6(2):41 –49, april-june 2007.
- [Res10] Nokia Research. Nokia sensor planet. <http://research.nokia.com/page/232>, 2010. Última consulta a 12 de Abril de 2012.

- [Ret06] G. Retscher. Location determination in indoor environments for location determination in indoor environments for location determination in indoor environments for pedestrian navigation. *Position, Location, And Navigation Symposium, 2006 IEEE/ION*, pages 547–555, April 2006.
- [RMI07] Jun Rekimoto, Takashi Miyaki, and Takaaki Ishizawa. Lifetag: Wifi-based continuous location logging for life pattern analysis. *LoCA*, 4718:35–49, 2007.
- [Rou07] Margaret Rouse. Web services (application services). <http://searchsoa.techtarget.com/definition/Web-services>, 2007. Última consulta a 15 de Agosto de 2012.
- [Roz10] Mike Rozlog. REST and SOAP: When Should I Use Each (or Both)? <http://www.infoq.com/articles/rest-soap-when-to-use-each>, Abril 2010. Última consulta a 15 de Agosto de 2012.
- [SB12] Markus Schief and Peter Buxmann. Business models in the software industry. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 3328 –3337, jan. 2012.
- [Sch03] Jochen Schiller. *Mobile Communications*. Addison-Wesley, 2nd edition edition, 2003.
- [Sci12] Citizen Science. Citizen science: Enabling participatory urbanism. <http://www.urban-atmospheres.net/CitizenScience/>, 2012. Última consulta a 10 de Abril de 2012.
- [SCP⁺11] Minh Shin, Cory Cornelius, Dan Peebles, Apu Kapadia, David Kotz, and Nikos Triandopoulos. AnonySense: A system for anonymous opportunistic sensing. *Journal of Pervasive and Mobile Computing*, 7(1):16–30, February 2011.
- [SGR02] K.V.S.S.S.S. Sairam, N. Gunasekaran, and S.R. Redd. Bluetooth in wireless communication. *Communications Magazine, IEEE*, 40(6):90 –96, jun 2002.
- [Sha07] Stephen Shankland. Google’s android parts ways with java industry group. http://news.cnet.com/8301-13580_3-9815495-39.html, Novembro 2007. Última consulta a 15 de Agosto de 2012.

- [Sha09] Jeffrey Sharkey. Google IO 09: Coding for Life – Battery Life, That Is. <http://www.google.com/events/io/2009/sessions/CodingLifeBatteryLife.html>, 2009. Última consulta a 04 de Setembro de 2012.
- [SHB⁺06] Mani Srivastava, Mark Hansen, Jeff Burke, Andrew Parker, and Sasank Reddy. Wireless urban sensing systems. Technical report, 2006.
- [SLB⁺03] Bill N. Schilit, Anthony LaMarca, Gaetano Borriello, William G. Griswold, David W. McDonald, Edward D. Lazowska, Anand Balachandran, Jason I. Hong, and Vaughn Iverson. Challenge: ubiquitous location-aware computing and the "place lab" initiative. In Parviz Kermani, editor, *WMASH*, pages 29–35. ACM, 2003.
- [Smi12] Chris Smith. Google play: 20 billion app downloads, 600,000 apps and games. <http://www.androidauthority.com/google-play-20-billion-app-downloads-600000-apps-games-98006/>, 2012. Última consulta a 15 de Agosto de 2012.
- [soa12] w3schools: SOAP Introduction. http://www.w3schools.com/soap/soap_intro.asp, 2012. Última consulta a 15 de Agosto de 2012.
- [SPG] U. Hengartner S. Pidcock, R. Smits and I. Goldberg. Notisense: An urban notification system to improve bystander privacy. *University of Waterloo*.
- [STT⁺08] T. Suzumura, S. Trent, M. Tatsubori, A. Tozawa, and T. Onodera. Performance comparison of web service engines in php, java and c. In *Web Services, 2008. ICWS '08. IEEE International Conference on*, pages 385–392, sept. 2008.
- [sum12] Projeto SUM: Sensing and Understanding human Motion dynamics. <http://www.humandigitalsystems.org/projects>, 2012. Última consulta a 15 de Agosto de 2012.
- [tec12] Techtracer: JSON Web Services – the XML JSON debate further ahead. <http://techtracer.com/2007/04/01/json-web-services-the-xml-json-debate-further-ahead/>, 2012. Última consulta a 15 de Agosto de 2012.

- [The11] The Financial Times. Apple and Google in Christmas showdownr. <http://www.ft.com/cms/s/2/6ecd4a0a-280a-11e1-a4c4-00144feabdc0.html#axzz22yG8POMC>, Dezembro 2011. Última consulta a 08 de Agosto de 2012.
- [TKFH06] Karen P. Tang, Pedram Keyani, James Fogarty, and Jason I. Hong. Putting people in their place: an anonymous and privacy-sensitive approach to collecting sensed data in location-based applications. In *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI*, pages 93–102, 2006.
- [TMN12] TMN. TMN: Loja Smartphones. <http://www.tmn.pt/portal/site/loja/menuitem.5608f05ec0c0f80726758710851056a0/?vgnextoid=180f4060d7dc7210VgnVCM1000005401650aRCRD>, 2012. Última consulta a 15 de Agosto de 2012.
- [urb10] Urban score: Measuring your relationship with the city. <http://www.urban-atmospheres.net/Experiments/UrbanScore/index.html>, 2010. Última consulta a 10 de Abril de 2012.
- [urb12] Urban atmospheres. <http://www.urban-atmospheres.net/>, 2012. Última consulta a 10 de Abril de 2012.
- [Vin10] Vinay. Goodbye SOAP – Welcome JSON REST. <http://www.skill-guru.com/blog/2010/11/02/goodbye-soap-welcome-json-rest/>, Novembro 2010. Última consulta a 15 de Agosto de 2012.
- [w3s12a] w3schools. w3schools: Introduction to XML. http://www.w3schools.com/xml/xml_what_is.asp, 2012. Última consulta a 15 de Agosto de 2012.
- [w3s12b] w3schools. w3schools: JSON Tutorial. <http://www.w3schools.com/json/default.asp>, 2012. Última consulta a 15 de Agosto de 2012.
- [W3T12] W3Techs. W3Techs - Web Technology Surveys: Usage statistics and market share of PHP for websites. <http://w3techs.com/technologies/details/pl-php/all/all>, Agosto 2012. Última consulta a 30 de Agosto de 2012.

- [WABS09] C. Weinhardt, A. Anandasivam, B. Blau, and J. Stosser. Business models in the service world. *IT Professional*, 11(2):28 –33, march-april 2009.
- [WMK11] A. Wakabayashi, S. Motomura, and S. Kato. Communicative humanoid robot control system reflecting human body movement. In *Micro-NanoMechatronics and Human Science (MHS), 2011 International Symposium on*, pages 122 –127, nov. 2011.
- [YCL11] Qilong Yuan, I-Ming Chen, and Shang Ping Lee. Slac: 3d localization of human based on kinetic human movement capture. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 848 –853, may 2011.
- [YDNY09] Zheng Yan, Yan Dong, Valtteri Niemi, and Guoliang Yu. Trust behaviors of mobile application usage - a formal experiment report. *Nokia Research Center - Technical Report*, August 25 2009.
- [YHSG03] Changqing Yin, Shaoyin Huang, Pengcheng Su, and Chuanshan Gao. Secure routing for large-scale wireless sensor networks. In *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, volume 2, pages 1282 – 1286 vol.2, april 2003.
- [Zhu03] Sencun Zhu. Leap: Efficient security mechanisms for large-scale distributed sensor networks. pages 62–72. ACM Press, 2003.
- [ZMvdB05] T. Zhang, S. Madhani, and E. van den Berg. Sensors on patrol (sop): using mobile sensors to detect potential airborne nuclear, biological, and chemical attacks. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 2924 –2929 Vol. 5, oct. 2005.

Apêndices

Apêndice A

Principais Tecnologias Disponíveis nos *Smartphones* – Explicação

0.1 Wi-Fi – IEEE 802.11

O Wi-Fi é uma tecnologia que permite a criação de *Wireless Local Area Network* (WLAN), possibilitando a ligação à rede sem fios através de ondas rádio. É baseada no *standard* IEEE 802.11 e usa as bandas de frequência dos 2.4 a 5 GHz, bandas estas atribuídas em 1985 pelo FCC para o uso industrial, científico e médico (ISM). A possibilidade de utilizar estas bandas de frequências possibilitou o aparecimento de tecnologias sem fios sem os custos de licenciamento. No início de 1990, o IEEE estabeleceu um comité executivo, como parte do grupo IEEE 802 para LAN e MAN, para se concentrar no desenvolvimento de uma norma de WLAN, chamado 802.11. Sendo um *standard* do grupo 802.x, significa que o padrão especifica a camada física e de acesso ao meio, adaptadas aos requisitos específicos das redes sem fios, mas mantém o mesmo interface para as camadas superiores como as restantes redes, de modo a manter a interoperabilidade [Sch03].

O objetivo primário do *standard* foi a especificação de uma WLAN única e robusta que permitisse serviços assíncronos e *time-bounded*. A camada MAC deveria ser capaz de operar com múltiplas camadas físicas, cada uma das quais exibindo diferentes características de transmissão.

As camadas da especificação IEEE802.11 são a camada de ligação e a camada física. A camada de ligação inclui o *Logical Link Control* (LLC) que aborda as diferenças do controlo das camadas de acesso ao meio, e o *Medium Access Control* (MAC) que controla o acesso ao meio e disponibiliza o suporte para *roaming*, autenticação e gestão de energia. A camada física é definida em três tipos *InfraRed* (IR), *Frequency-Hopping Spread-Spectrum* (FHSS) e *Direct Sequence Spread Spectrum* (DSSS) [Sch03] dependendo do acesso físico.

As redes Wi-Fi podem ter duas arquiteturas distintas, baseadas em ligações *ad-hoc*, por

exemplo entre dois computadores, ou baseadas em infraestrutura, como a representada na figura A.1. Na figura podem ser observados vários dispositivos (*STations* (STA)) conectados aos pontos de acesso (*Access Points* (AP)). As STAs são dispositivos com capacidade para a interface de rádio com as redes Wi-Fi, que juntamente com o AP dentro da mesma cobertura rádio formam um *Basic Service Set* (BSS). No exemplo são apresentados dois BSSs que se encontram ligados através do sistema de distribuição (*Distribution System* (DS)). Um DS liga vários BSSs através de APs de modo a formar um única rede e a estender a área de cobertura da rede, passando esta a chamar-se *Extended Service Set* (ESS) com o seu próprio identificador, o ESSID. O ESSID é o nome da rede, sendo utilizado para separar diferentes redes. O DS liga a rede sem fios a um portal que permite a ligação a outras redes LAN.

O DS lida com a transferência de dados entre diferentes APs. Os APs fornecem sincronização dentro de um BSS, suportando gestão de energia e podem controlar o acesso ao meio para suportar serviços *time-bounded*.

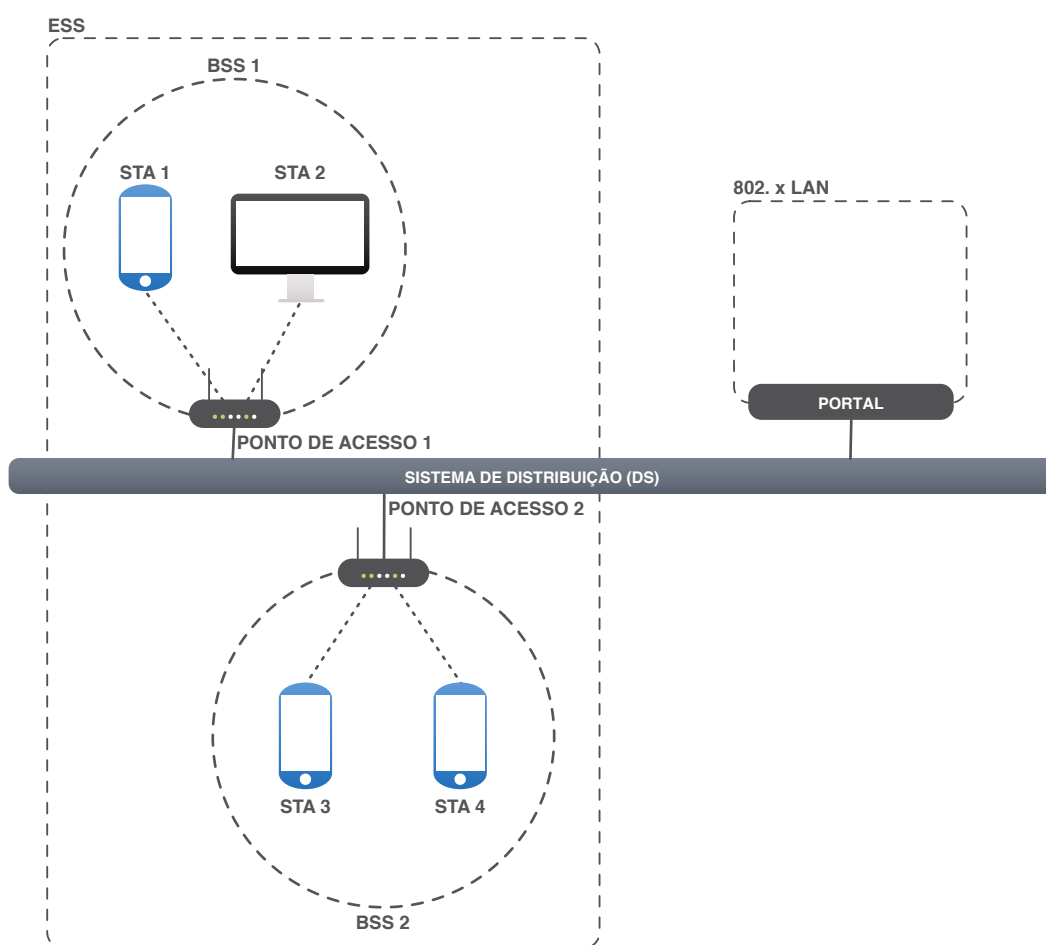


Figura A.1: Arquitetura de uma rede Wi-Fi (IEEE802.11) baseada em infraestrutura[Sch03]

A primeira versão do padrão foi aprovada em 1997, com taxas de transmissão de dados de 2 Mb/s. Uma grande parte do sucesso do 802.11 foi devida à sua compatibilidade com

redes Ethernet. Foram criados posteriormente novos grupos de trabalho, com o objetivo de melhorar a norma existente. O grupo de trabalho 802.11b criou um novo padrão, adotado pelo IEEE em 1999, permitindo taxas de transmissão de dados de 5.5 e 11 Mb/s, com alcance que pode atingir os 300 metros em espaço livre utilizando acesso ao meio baseado em DSSS e funcionando na banda dos 2.4 GHz. O grupo 802.11a criou um padrão baseado em modulação *Orthogonal Frequency-Division Multiple Access* (OFDMA) que permite velocidades até 54 Mb/s funcionando na banda dos 5 GHz.

Foi posteriormente criado o grupo de trabalho 802.11g, responsável por utilizar a modulação OFDMA na banda dos 2.4 GHz, criando um novo padrão compatível com o existente 802.11b. Em setembro de 2003 foi criado o grupo 802.11n, que conseguiu exceder os 54 Mb/s utilizando MIMO como técnica de transmissão; em 2009 o padrão 802.11n conseguiu taxas de transmissão até 600 Mb/s mas, na sua atual implementação, devido ao congestionamento na banda de 2.4 GHz, o máximo são 104 Mb/s[Sch03].

Outro padrão em desenvolvimento é o 802.11ac, que funcionará na banda de 5 GHz e que permitirá transferências multi-estação WLAN, atingindo teoricamente taxas de pelo menos 1 Gb/s e uma taxa máxima através de um único elo de pelo menos 500 Mb/s. Isto é conseguido através da extensão dos conceitos de interface do 802.11n e da utilização de mais fluxos MIMO(até 8), de multi-utilizador MIMO e modulação de elevada ordem (até 256 QAM), possibilitando o funcionamento de aplicações como a transmissão simultânea de vídeo HD para vários clientes pela casa, e sincronização e backup de grandes quantidades de dados rapidamente.

0.2 Bluetooth

O Bluetooth é uma tecnologia sem fios de baixo custo, para ligações de curto alcance entre dispositivos. Foi desenvolvida pela Ericson em 1994 com o objetivo de se tornar uma alternativa aos cabos de ligação[MW05], começando a ser incorporada numa grande variedade de dispositivos como PDAs, telemóveis, computadores e periféricos, como ratos, teclados, impressoras, e auriculares.

Contudo, as taxas de transmissão do Bluetooth eram baixas, suportando na versão 1.2 apenas 780 Kb/s, dos quais 721 Kb/s eram utilizados para transferência de dados num único sentido (57.6 Kb/s de retorno) ou a transferência de dados simétrica até 432.6 Kb/s. Mostrou-se por isso inadequado para a transferências de grandes volumes de dados mas suficiente para aplicações de impressão e de voz, sendo o seu principal ponto forte a capacidade para simultaneamente lidar com transmissões de dados e de voz.

O Bluetooth, é capaz de suportar um canal de dados assíncrono e até três canais de voz

síncronos, ou um canal suportando voz e dados simultaneamente.[SGR02]. Os dispositivos Bluetooth têm diferentes alcances dependendo da sua classe. A classe 1 são, por exemplo, os módulos amovíveis ou internos para computadores, e podem atingir os 100 metros; a classe 2, por exemplo, os módulos nos *smartphones* e telemóveis com alcance até 10 metros; e a classe 3, auriculares e *headphones*, com alcance até 1 metro. O que caracteriza cada classe é a potência transmitida.

Em 1998 a Intel, a IBM, a Nokia e a Toshiba, juntaram-se à Ericson formando o Bluetooth *Special Interest Group* (SIG), e em 1999 juntaram-se mais empresas, incluindo a Microsoft e a Motorola. O trabalho do SIG permitiu a evolução do Bluetooth para padrões abertos de modo a garantir a rápida aceitação no mercado. A *Wireless Personal Area Network* (WPAN) que é baseada na especificação do Bluetooth, é hoje uma norma do IEEE (IEEE 802.15.1).

A especificação do Bluetooth define como os dispositivos Bluetooth se agrupam para fins de comunicação. Uma Bluetooth *Wireless Personal Area Network* (BT-WPAN) consiste em *piconets*, que são formadas por um conjunto de até oito dispositivos Bluetooth. Numa *piconet* um dispositivo é designado como mestre e os restantes são escravos. Duas *piconets* podem ser ligadas através de um dispositivo Bluetooth comum de modo a formar uma *scatternet*. Estas *piconets* interligadas dentro da *scatternet* formam um *backbone* para a *Mobile Area Network*(MANET), o qual permite ligar dispositivos que não têm comunicação direta entre si, ou que estão fora do alcance de outro dispositivo, para trocar dados através de vários saltos na *scatternet*[MW05]. As atuais implementações Bluetooth são primariamente ligações de dados simples ponto-a-ponto entre dispositivos Bluetooth em alcance direto.

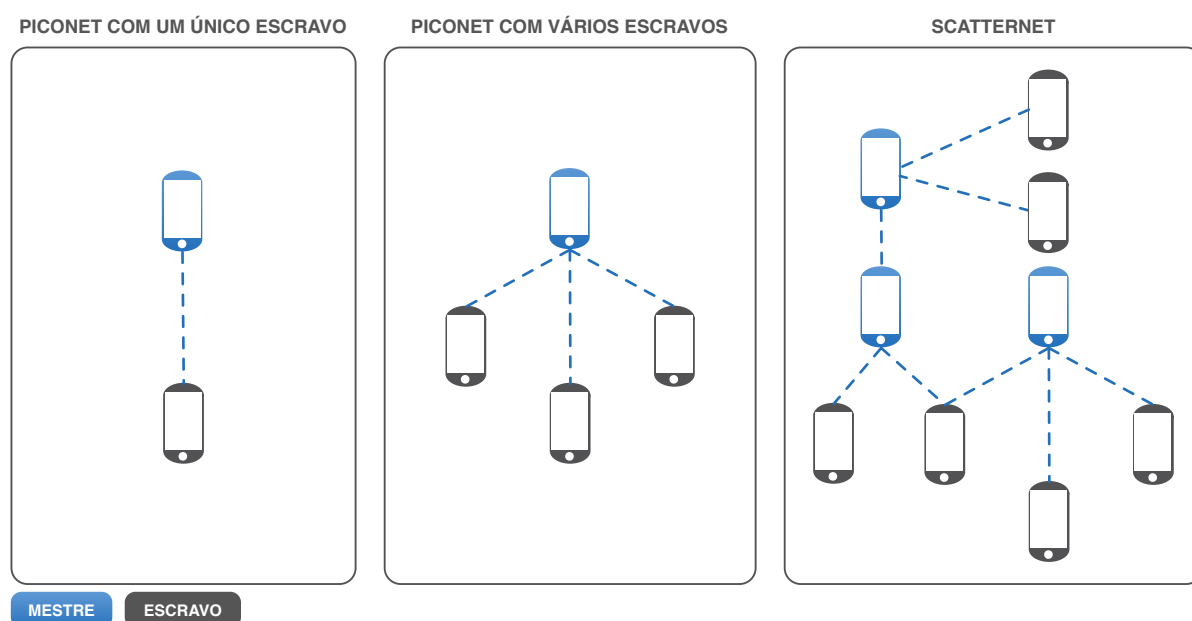


Figura A.2: Bluetooth *piconets* e *scatternets*[Por12]

A especificação do Bluetooth divide a pilha protocolar em três grupos lógicos: O grupo de protocolos de transporte, o grupo de protocolos de *Middleware* e o grupo de aplicação. O grupo de protocolos de transporte permite que os dispositivos Bluetooth se localizem uns aos outros e façam a gestão das ligações físicas e lógicas com os protocolos da camada superior e com as aplicações. O grupo protocolar *Middleware*, inclui os protocolos padrão de indústria e de terceiros, assim como os protocolos desenvolvidos pelo SIG. Estes protocolos permitem que as aplicações operem sobre as ligações Bluetooth. O grupo de aplicação consiste em aplicações que utilizam as ligações Bluetooth[MW05][Por12].

Um transdutor Bluetooth é um dispositivo baseado em FHSS que utiliza a banda de frequências ISM dos 2.4 GHz. Na maioria dos países estão disponíveis 79 canais, contudo alguns países autorizam apenas a utilização de 23 canais. A largura de banda normal de cada canal é de 1MHz.

A especificação Bluetooth utiliza *Time Division Duplexing* (TDD) e *Time Division Multiple Access* (TDMA) e modulação *Gaussian Frequency Shift Keying* (GFSK), com débitos de 1 Mb/s em modo *Basic Rate* (BR) e de 2 ou 3 Mb/s em modo *Enhanced Data Rate* (EDR). A versão atual do Bluetooth é a 4.0, adotada em 2010.

0.3 Rede Móvel Celular

As redes móveis celulares digitais são extensões das tradicionais redes PSTN e ISDN, e permitem o *roaming* entre redes de diferentes países. Até há algum tempo, estes sistemas eram utilizados maioritariamente para tráfego de voz, no entanto o tráfego de dados está em constante crescimento, levando à evolução destas redes e ao aparecimento de novas soluções.

O sistema digital mais popular é o GSM, existindo ainda o sistema analógico AMPS, o Japonês PDC, o CDMA e TDMA. A Europa, praticamente toda, utiliza GSM, nos Estados Unidos é adotado o GSM, TDMA e CDMA.

■ 1G - *Advanced Mobile Phone Service* (AMPS)

A primeira geração (1G) era baseada em transmissões analógicas de voz. Um exemplo é a tecnologia AMPS que foi a pioneira e ajudou a impulsionar a utilização da tecnologia celular. No entanto tinha alguns problemas, não era cifrada e por isso vulnerável a escutas, era suscetível a clonagem dos dispositivos celulares e utilizava *Frequency-Division Multiple Access* (FDMA), necessitando de uma quantidade significativa de espectro para funcionar. Muitos dos primeiros telefones celulares comerciais como o Motorola *DynaTAC Analog* AMPS foram substituídos pelo aparecimento da Digital AMPS (D-AMPS) em 1990 e o serviço AMPS foi desligado pela

maioria das operadoras em 2008.

■ 2G – *Global System for Mobile Communications* (GSM)

No início dos anos oitenta a Europa tinha um grande número de sistemas móveis analógicos baseados em standards similares, mas que funcionavam em diferentes frequências. De modo a evitar esta situação e a obter-se uma segunda geração totalmente digital, foi fundado em 1982 o *Groupe Spécial Mobile* (GSM). Este sistema seria rapidamente nomeado de *Global System for Mobile Communications* (GSM), ficando o processo de especificação a cargo do ETSI. No contexto do UMTS e da criação do 3GPP (*Third Generation Partnership Project*) todo o processo de desenvolvimento do GSM foi transferido para o 3GPP e o futuro desenvolvimento combinado com o desenvolvimento do 3G.

O objetivo primário do GSM foi fornecer um sistema de comunicações móveis que permitisse aos utilizadores o *roaming* pela Europa, e que fosse compatível com os outros sistemas ISDN e PSTN. O GSM é um sistema tipicamente de segunda geração (2G), vindo substituir os sistemas analógicos de primeira geração, mas não oferecendo elevadas taxas de transmissão de dados. Foi inicialmente lançado na Europa utilizando as frequências 890–915 MHz para *uplink* e 935–960 MHz para *downlink*, sendo chamado de GSM 900 de modo a distingui-lo das versões que surgiram posteriormente, como o GSM 1800 também denominado de DCS (*Digital Cellular System*) 1800, e o sistema GSM 1900 maioritariamente utilizados nos Estados Unidos, também denominado de PCS (*Personal Communications Service*) 1900.

Em relação ao interface de rádio, o GSM utiliza vários mecanismos para *multiplexing* e acesso ao meio, implementando SDMA (*Space-Division Multiple Access*) utilizando células com *Base Transceiver Station* (BTS) e associando um MS (*Mobile Station*) a um BTS. É utilizado FDD (*Frequency-Division Duplexing*) para separar o *downlink* e o *uplink*.

Devido às baixas taxas de transferência de dados, foram desenvolvidos novos serviços para aumentar o débito, entre os quais o GPRS (*General Packet Radio Service*) e o EDGE (*Enhanced Data rates for GSM Evolution*), que permitiu aos sistemas GSM suportarem débitos superiores, com o EDGE a ir até aos 384 Kb/s.

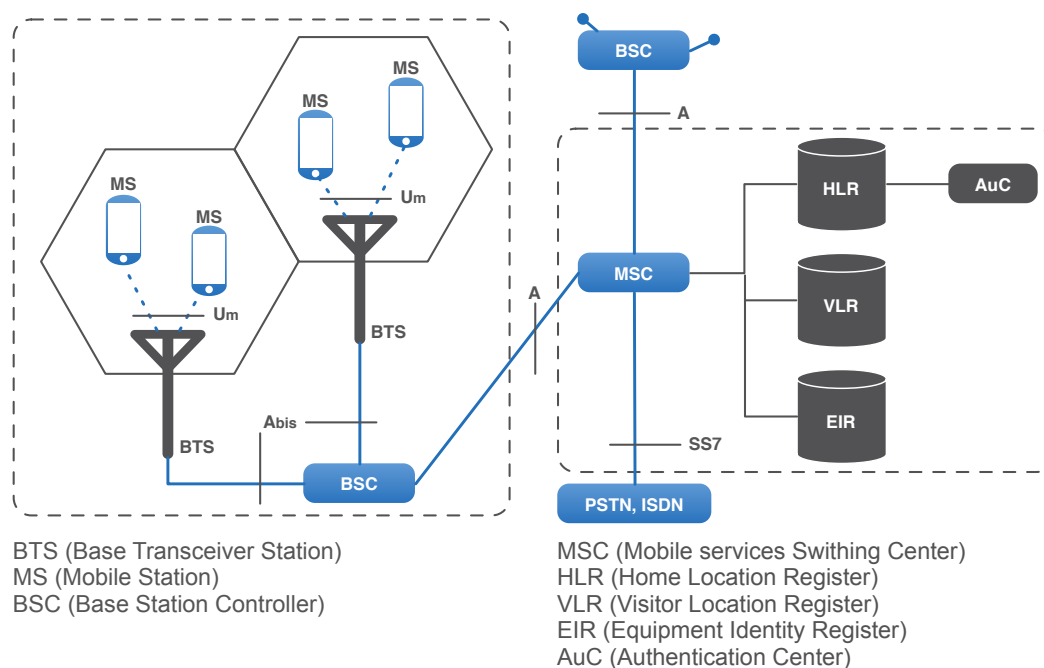


Figura A.3: Arquitetura do sistema GSM

■ 3G – *Universal Mobile Telecommunications System* (UMTS)

Com a difusão da utilização das redes 2G as pessoas começaram a utilizar mais os seus telemóveis no dia a dia, tornando-se evidente que a utilização de serviços de dados, como o acesso à Internet, estavam em crescimento. Além disto, o exemplo dos serviços de banda larga fixa, que cada vez ofereciam maiores débitos, vislumbrou que também nas redes móveis seria necessário evoluir. A tecnologia 2G não estava preparada para dar resposta a estes requisitos, por isso a indústria começou a trabalhar na próxima geração, conhecida como 3G.

A principal diferença tecnológica que distingue a terceira da segunda geração, é a completa alteração da transmissão de dados da comutação de circuitos para a comutação de pacotes. Além disso, o processo de normalização focou-se mais nos requisitos do que na tecnologia (por exemplo o requisito de 2 Mb/s máximos em ambientes fechados e 384 Kb/s em ambientes abertos).

O UMTS fornece diversas interfaces de radio terrestres, chamadas UMTS *Terrestrial Radio Access* (UTRA) sendo o UTRA-FDD o mais popular. O UTRA-FDD utiliza *Wideband Code Division Multiple Access* (W-CDMA) com *Direct Sequence Spreading* (DSS), de modo a oferecer maior eficiência espectral e maior largura de banda. Uma estação móvel (*Mobile Station* (MS)), na Europa utiliza uma portadora dentro da banda de 1920 e 1980 MHz para *uplink* e a estação base utiliza uma portadora dentro da banda de 2110 a 2170 MHz para *downlink*[Sch03].

A arquitetura da rede UTRA(Fig.A.4) consiste em vários *Radio Network Subsystems* (RNS).

Cada RNS é controlado por um *Radio Network Controller* (RNC) que liga a vários *Node B*. Um RNC em UMTS pode ser comparado a um BSC no GSM e o *Node B* é similar a um BTS. Cada *Node B* pode controlar várias antenas que formam uma célula de radio. A estação móvel pode estar ligada a uma ou mais células. Cada RNC está ligado à *Core Network* (CN).

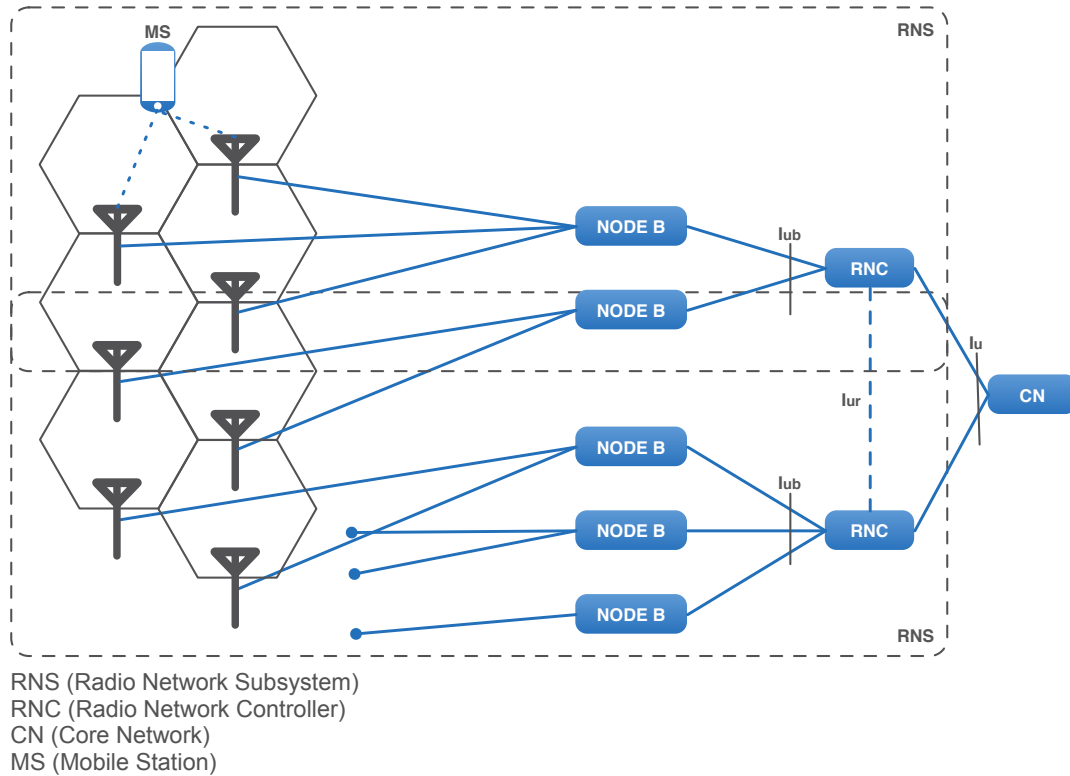


Figura A.4: Arquitetura básica de uma rede UTRAN[Sch03]

Em meados de dois mil começou a ser implementada uma evolução da tecnologia 3G, chamada *High-Speed Downlink Packet Access* (HSDPA) que permitiu que as redes baseadas em UMTS atingissem maiores capacidades e taxas de transferência de dados. As implementações atuais HSDPA permitem taxas de 1.8, 3.6, 7.2 e 14.0 Mb/s de *downlink*. Mais tarde surgiu o HSPA+ e passou a ser possível taxas até 21 Mb/s de *downlink*.

Com o recente aparecimento do *HSPA*¹ *Multicarrier Evolution*, teóricamente são possíveis taxas máximas de 168 Mb/s de *downlink* e 22 Mb/s de *uplink*. Os 168 Mb/s são obtidos utilizando múltiplas portadoras simultaneamente(neste caso quatro 4C-HSDPA), modulação 64-QAM e *Multiple-Input and Multiple-Output* (MIMO)[JBG⁺09][HT07]. Alguns dispositivos, como o iPhone 5[App12f] da Apple recentemente lançado em Setembro de 2012, suporta, além de LTE, *Dual Cell-High Speed Downlink Packet Access* (DC-HSDPA) sem MIMO, que atinge 42.2 Mb/s de *downlink* máximo[HT07].

¹*High Speed Packet Access* (HSPA) é a junção dos protocolos de *downlink* e *uplink*, respetivamente *High Speed Downlink Packet Access* (HSDPA) e *High Speed Uplink Packet Access* (HSUPA)

■ 4G - *Long Term Evolution* (LTE)

Long Term Evolution (LTE), no mercado conhecido como 4G LTE, é uma norma de comunicações sem fios que permite elevadas taxas na transferência de dados em dispositivos móveis e terminais. É baseada nas tecnologias de rede GSM/EDGE e UMTS/HSPA, aumentando a capacidade e a taxa através de um interface de radio diferente e melhoramentos do núcleo da rede. Apesar de no mercado esta tecnologia ser considerada de quarta geração, a norma LTE como está especificado na *Release* 8 e 9 do 3GPP não satisfaz os requisitos técnicos do consórcio para o padrão de nova geração, estabelecidos pela organização ITU-R. Contudo, devido às pressões do mercado e aos avanços significativos que o WIMAX, HSPA+ e o LTE permitiram em relação ao 3G original, foi decidido que estas tecnologias poderiam ser consideradas de quarta geração. O padrão LTE *Advanced* formalmente satisfaz os requisitos do ITU-R e é por isso considerado como o verdadeiro 4G.

O 3GPP começou a investigar o UTRAN LTE em 2004 definindo os requisitos do padrão, entre os quais se encontram a otimização da comutação de pacotes, redução do atraso no *round-trip time* do servidor para o equipamento do utilizador para valores a baixo dos 30 ms, tempo de acesso abaixo de 300 ms, taxas máximas de *uplink/downlink* de 50/100 Mb/s e bons níveis de mobilidade e segurança[HT07].

É utilizado *Orthogonal Frequency Division Multiple Access* (OFDMA) para o *downlink* e *Single Carrier Frequency Division Multiple Access* (SC-FDMA) para o *uplink*. Uma abordagem baseada em OFDMA já havia sido anteriormente considerada para o UMTS, no entanto nessa altura ainda não estava suficientemente desenvolvida. O OFDMA pode também trazer benefícios aos sistemas baseados em *Code Division Multiple Access* (CDMA) com o aumento da largura de banda, além disso o tratamento com diferentes larguras de banda no mesmo sistema é mais flexível com OFDMA.

Foi adotada uma arquitetura plana. A arquitetura inicial do UMTS foi definida como hierárquica, as funcionalidades relacionadas com o acesso radio estavam no *Radio Network Controller* (RNC), na arquitetura plana essas funcionalidades estão localizadas na estação base.

O LTE permite taxas máximas de transferência elevadas, utilizando maior largura de banda (até 20MHz), modulação 64-QAM e transmissões *multistream* MIMO. As taxas máximas de transferência de dados teóricas são de 170 Mb/s utilizando 2X2 MIMO e de 340 Mb/s utilizando 4X4 MIMO, enquanto a taxa de *upload* pode atingir os 86 Mb/s utilizando 64-QAM. As metas de 100 Mb/s de *downlink* e de 50 Mb/s de *uplink* do LTE foram claramente atingidas[HT07].

0.4 *Global Positioning System* (GPS)

O GPS é um sistema de navegação baseado em satélites desenvolvido pelo departamento da defesa dos Estados Unidos no início dos anos 70. Inicialmente o GPS era apenas utilizado em contexto militar, sendo mais tarde disponibilizado para utilização civil. O sistema é mantido pelo governo dos Estados Unidos e está acessível gratuitamente a qualquer pessoa que possua um recetor GPS. O GPS fornece informações de posicionamento e horário continuamente em qualquer parte do mundo e em qualquer situação meteorológica. Por servir um número ilimitado de utilizadores, e por razões de segurança, o GPS é um sistema passivo de funcionamento unidirecional (*one-way-ranging*)[ER06].

O sistema GPS consiste numa constelação de 24 satélites operacionais. Esta constelação, conhecida como *Initial Operational Capability* (IOC) ficou completa em Julho de 1993. De modo a garantir a total e constante cobertura em todo o mundo, os satélites GPS estão dispostos de modo a que quatro satélites estejam em cada um dos seis planos orbitais. Com esta geometria da constelação, 4 satélites GPS estão sempre visíveis em qualquer parte do mundo se for considerado um ângulo de elevação de 10° . Apenas quatro satélites são necessários para fornecer as informações de posicionamento ou de localização[ER06].

As órbitas dos satélites GPS são quase circulares e com uma inclinação de cerca de 55° relativamente ao plano do equador. O semieixo maior de uma órbita GPS é de cerca de 26.560 km, sendo o período de orbita GPS é cerca de 12 horas. O sistema foi declarado como estando plenamente operacional a 17 de Julho de 1995, garantindo a disponibilidade de pelo menos 24 satélites operacionais. Na verdade, uma vez atingidos os 24 satélites o número de satélites da constelação GPS foi sempre maior do que 24[ER06].

A ideia por detrás do GPS é relativamente simples. Se forem conhecidas as distancias de um ponto na terra (um recetor de GPS), a três satélites GPS juntamente coma localização destes, então a localização do recetor pode ser determinada aplicando o conceito de *Resection*[ER06]. Cada satélite transmite constantemente uma mensagem que inclui a hora de transmissão da mensagem e a posição do satélite à hora da transmissão.

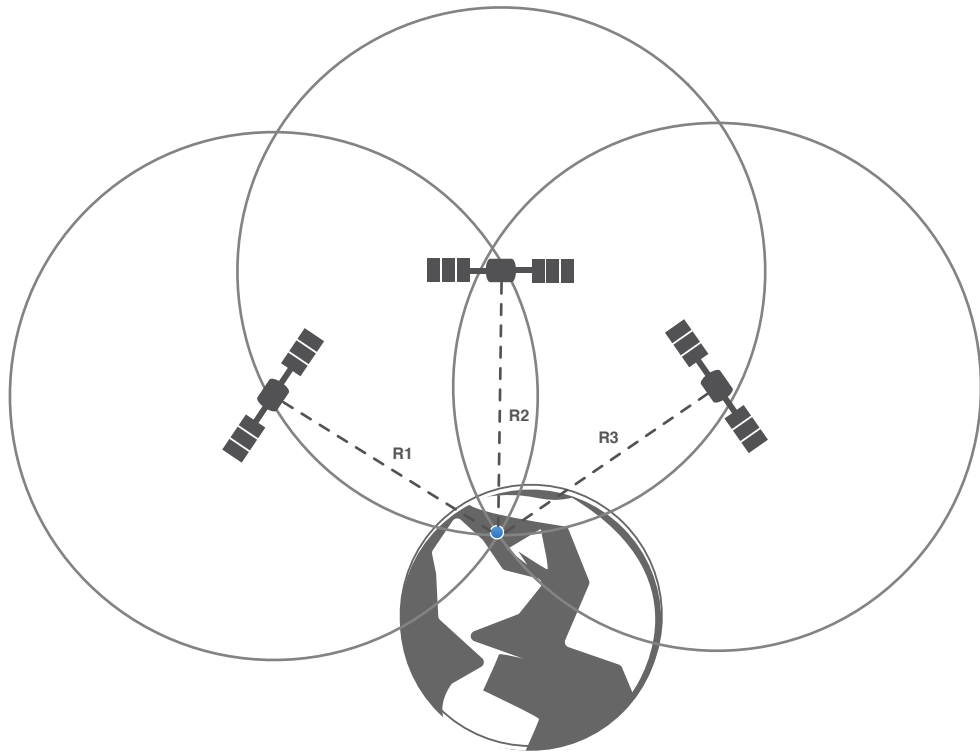


Figura A.5: Posicionamento GPS

Teóricamente, apenas três distancias para três satélites (detetados simultaneamente) são necessárias. Neste caso, o recetor estará localizado na intersecção das três esferas, cada uma com centro no satélite e raio igual à distancia até ao recetor. No entanto, do ponto de vista prático, um quarto satélite é necessário para compensar o desfasamento do relógio do recetor. O recetor GPS utiliza estas informações para calcular coordenadas geográficas (latitude e longitude), a altitude e a velocidade de deslocação do utilizador.

A precisão da posição GPS é aproximadamente de 22 m. Para melhorar a precisão é utilizado o método diferencial, que utiliza dois recetores em simultâneo que seguem a posição dos mesmos satélites GPS. Neste caso podem ser obtidos níveis de precisão na ordem de alguns centímetros a poucos metros [ER06].

Além do GPS, outros sistemas estão em utilização em desenvolvimento. O sistema russo *GLObal NAVigation Satellite System* (GLONASS) esteve em utilização apenas pelos militares russos, ficando disponível para os civis em 2007. Um dos sistemas em atual desenvolvimento é o sistema da União Europeia Galileo. O desenvolvimento e disponibilização de novos sistemas de posicionamento permitirá que os utilizadores não fiquem totalmente dependentes do sistema GPS.

Apêndice B

Diagramas de Classes

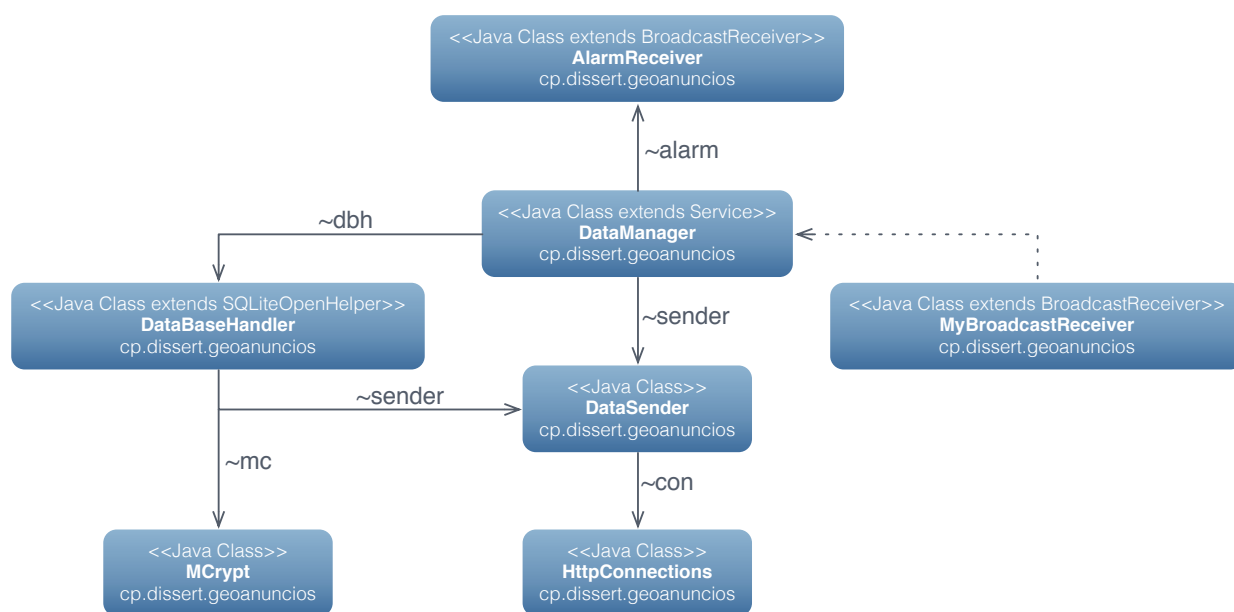


Figura B.1: Módulo de Recolha de Dados

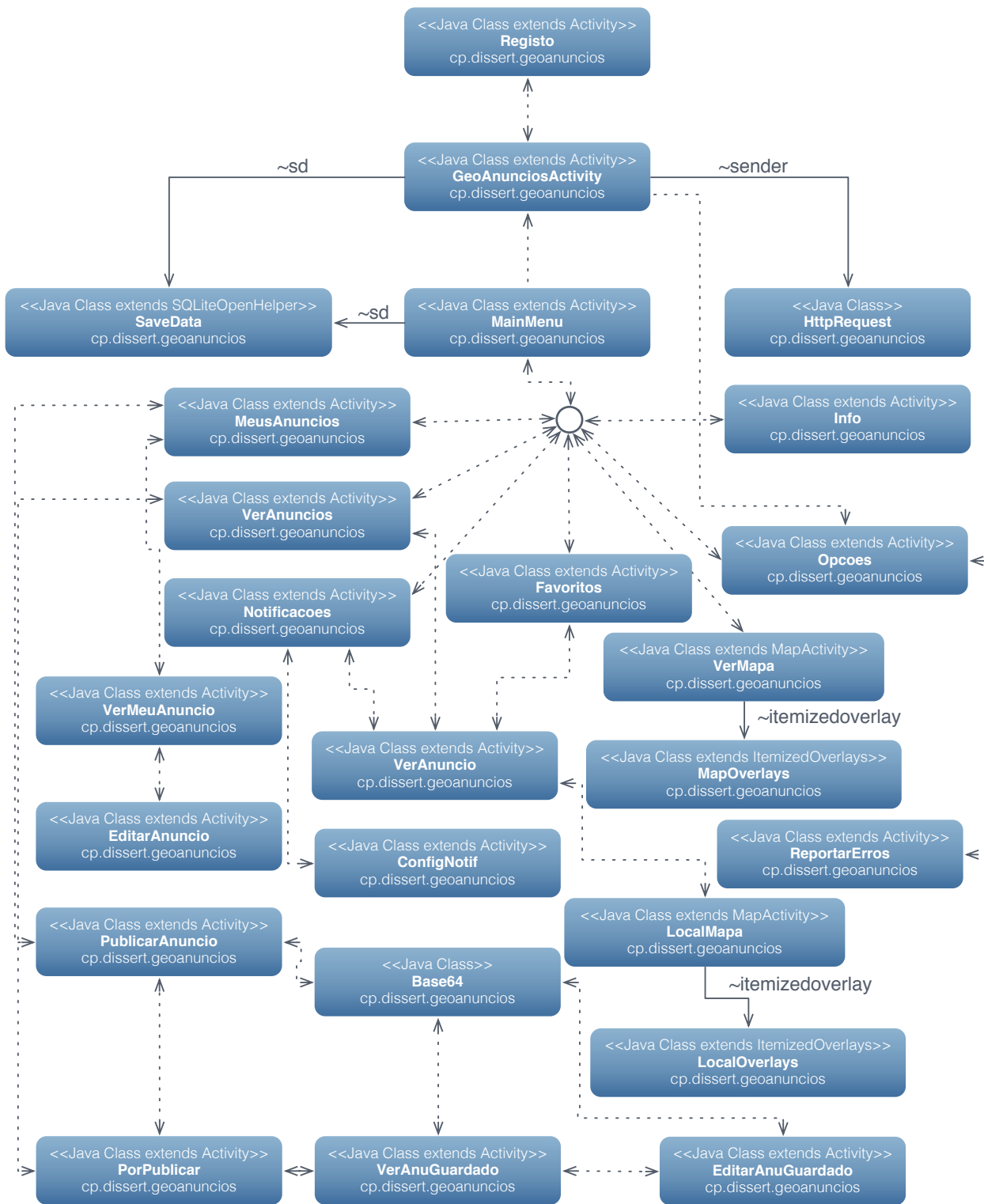


Figura B.2: GeoAnúncios

Apêndice C

Protocolo de Comunicação

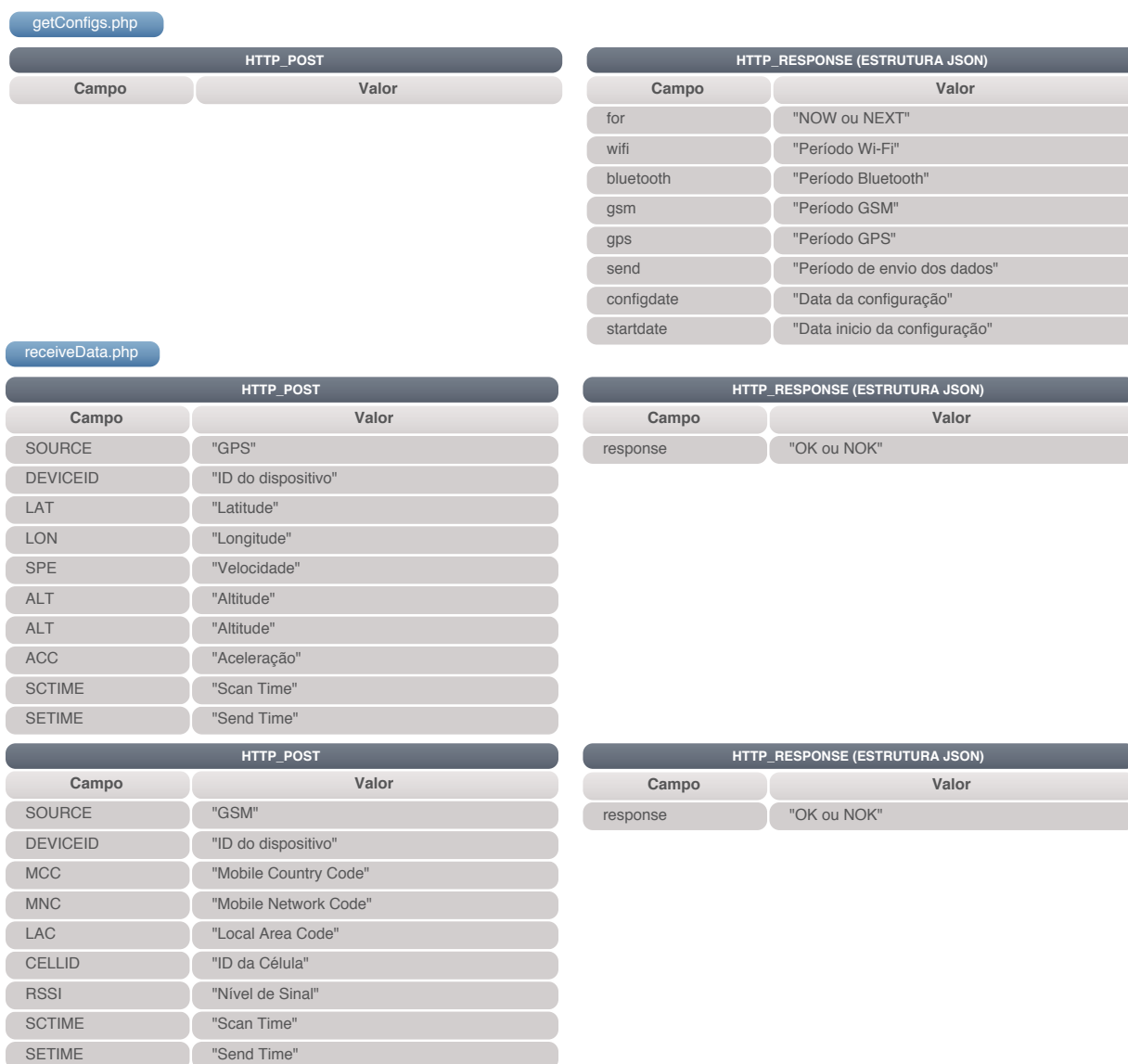


Figura C.1: Protocolo – Recolha de Dados (1)

HTTP_POST		HTTP_RESPONSE (ESTRUTURA JSON)	
Campo	Valor	Campo	Valor
SOURCE	"WIFI"	response	"OK ou NOK"
DEVICEID	"ID do dispositivo"		
MCC	"Mobile Country Code"		
MNC	"Mobile Network Code"		
LAC	"Local Area Code"		
CELLID	"ID da Célula"		
RSSI	"Nível de Sinal"		
SCTIME	"Scan Time"		
SETIME	"Send Time"		

HTTP_POST		HTTP_RESPONSE (ESTRUTURA JSON)	
Campo	Valor	Campo	Valor
SOURCE	"GSM"	response	"OK ou NOK"
DEVICEID	"ID do dispositivo"		
DEVICEMAC	"Mac Address do interface Wi-Fi"		
BSSID	"Mac Adress Ponto de Acesso"		
SSID	"Nome do Ponto de Acesso"		
RSSI	"Nível de Sinal"		
CHANNEL	"Frequência"		
CAPABILITIES	"Capacidades do Ponto de Acesso"		
SCTIME	"Scan Time"		
SETIME	"Send Time"		

HTTP_POST		HTTP_RESPONSE (ESTRUTURA JSON)	
Campo	Valor	Campo	Valor
SOURCE	"BLUETOOTH"	response	"OK ou NOK"
DEVICEID	"ID do dispositivo"		
DEVICEMAC	"Mac Address do interface Bluetooth"		
MAC	"Mac Adress do dispositivo"		
NAME	"Nome do dispositivo"		
CLASS	"Classe do dispositivo"		
RSSI	"Nível de Sinal"		
SERVICES	"Serviços Suportados"		
SCTIME	"Scan Time"		
SETIME	"Send Time"		

HTTP_POST		HTTP_RESPONSE (ESTRUTURA JSON)	
Campo	Valor	Campo	Valor
SOURCE	"BATTERY"	response	"OK ou NOK"
DEVICEID	"ID do dispositivo"		
STIME	"Inicio da Recolha de dados"		
TIME	"Data e hora atual"		
LEVEL	"Nível de bateria"		
SETIME	"Send Time"		

Figura C.2: Protocolo – Recolha de Dados (2)

regUser.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
Email	"Email"	response	"OK ou NOK"
Password	"Password"		
login.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
Email	"Email"	response	"OK ou NOK"
Password	"Password"		
checkDeviceID.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
DeviceID	"ID do dispositivo"	response	"OK (Associado à conta atual ou AA (Já associado a outra conta) ou NA (Ainda não associado))"
regDevice.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
DeviceID	"ID do dispositivo"	response	"OK ou NOK"
updateDeviceReg.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
DeviceID	"ID do dispositivo"	response	"OK ou NOK"
checkAppVersion.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
		version	"Versão atual da aplicação"
		releaseDate	"Data da versão"
		notes	"Notas sobre a versão"
getUserInfo.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
		Email	"Email do utilizador autenticado"
		Credits	"Créditos do utilizador"
repPass.php - Recuperar password		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
Email	"Email introduzido"	response	"OK ou NOK"
getNotifications.php		HTTP_RESPONSE (ESTRUTURA JSON com 1 ou N subestruturas)	
HTTP_POST		Campo Valor	
filter	"Filtro de categorias"	Adid	"ID do anúncio"
lctime	"Data e hora da última verificação"	Category	"Categoria"
sendtime	"Data e hora do pedido para acerto"	Title	"Título"
		Description	"Descrição"
		Lat	"Latitude"
		Lon	"Longitude"
		Address	"Endereço"
		PubDate	"Data de publicação"
getNotifications.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo Valor	
Email	"Email do utilizador"	response	"OK ou NOK"
Resposta	"Resposta ao anúncio"		
Adid	"ID do anúncio"		

Figura C.3: Protocolo – GeoAnúncios (1)

getUserAds.php		HTTP_RESPONSE (ESTRUTURA JSON com 1 ou N subestruturas)	
HTTP_POST		Campo	
Campo	Valor	Campo	Valor
		Adid	"ID do anúncio"
		Category	"Categoria"
		Title	"Título"
		Description	"Descrição"
		Lat	"Latitude"
		Lon	"Longitude"
		Address	"Endereço"
		PubDate	"Data de publicação"
		ExpDate	"Data de Expiração"

getCategories.php		HTTP_RESPONSE (ESTRUTURA JSON com 1 ou N subestruturas)	
HTTP_POST		Campo	
Campo	Valor	Campo	Valor
		Name	"Nome da categoria"
		Description	"Descrição da categoria"

getAds.php		HTTP_RESPONSE (ESTRUTURA JSON com 1 ou N subestruturas)	
HTTP_POST		Campo	
Campo	Valor	Campo	Valor
		Category	"Categoria"
		Title	"Título"
		Description	"Descrição"
		Lat	"Latitude"
		Lon	"Longitude"
		Address	"Endereço"
		PubDate	"Data de publicação"

getAdDetails.php		HTTP_RESPONSE (ESTRUTURA JSON com 1 ou N subestruturas)	
HTTP_POST		Campo	
Campo	Valor	Campo	Valor
Adid	"ID do dispositivo"	Adid	"ID do anúncio"
		Category	"Categoria"
		Title	"Título"
		Description	"Descrição"
		Lat	"Latitude"
		Lon	"Longitude"
		Address	"Endereço"
		PubDate	"Data de publicação"
		ExpDate	"Data de Expiração"

editAd.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo	
Campo	Valor	Campo	Valor
adid	"ID do anúncio"	response	"OK ou NOK"
category	"Categoria"		
title	"Título"		
description	"Descrição"		
latitude	"Latitude"		
longitude	"Longitude"		
expdate	"Data de Expiração"		

deleteAd.php		HTTP_RESPONSE (ESTRUTURA JSONs)	
HTTP_POST		Campo	
Campo	Valor	Campo	Valor
Adid	"ID do anúncio"	response	"OK ou NOK"

Figura C.4: Protocolo – GeoAnúncios (2)

changePass.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo	
Campo	Valor	Campo	
newpass	"Nova Password"	response	"OK ou NOK"

pubAd.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo	
Campo	Valor	Campo	
category	"Categoria"	response	"OK ou NOK"
title	"Titulo"		
description	"Descrição"		
latitude	"Latitude"		
longitude	"Longitude"		
Address	"Endereço"		
expdate	"Data de Expiração"		
image	"Imagem: Base64 encoded String"		

searchAds.php		HTTP_RESPONSE (ESTRUTURA JSON com 1 ou N subestruturas)	
HTTP_POST		Campo	
Campo	Valor	Campo	
category	"Categoria a filtrar ou Todas"	Adid	"ID do anúncio"
toggles	"content ou city ou both"	Category	"Categoria"
query	"Texto a procurar"	Title	"Titulo"
		Description	"Descrição"
		Lat	"Latitude"
		Lon	"Longitude"
		Address	"Endereço"
		PubDate	"Data de publicação"

logout.php		HTTP_RESPONSE (ESTRUTURA JSON)	
HTTP_POST		Campo	
Campo	Valor	Campo	
		response	"OK ou NOK"

Figura C.5: Protocolo – GeoAnúncios (3)

Apêndice D

Aplicação GeoAnúncios

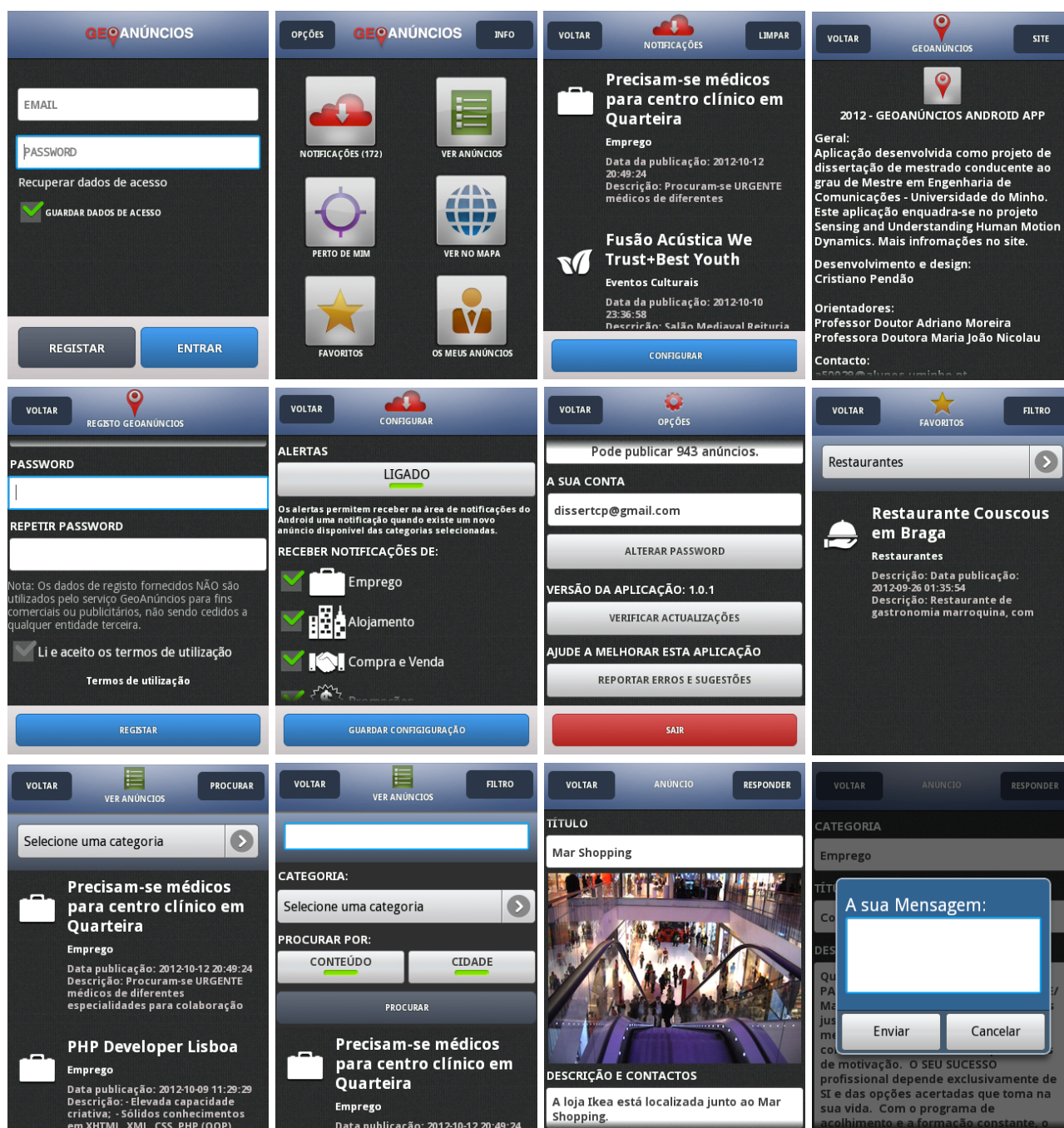


Figura D.1: Ecrãs da Aplicação (1)

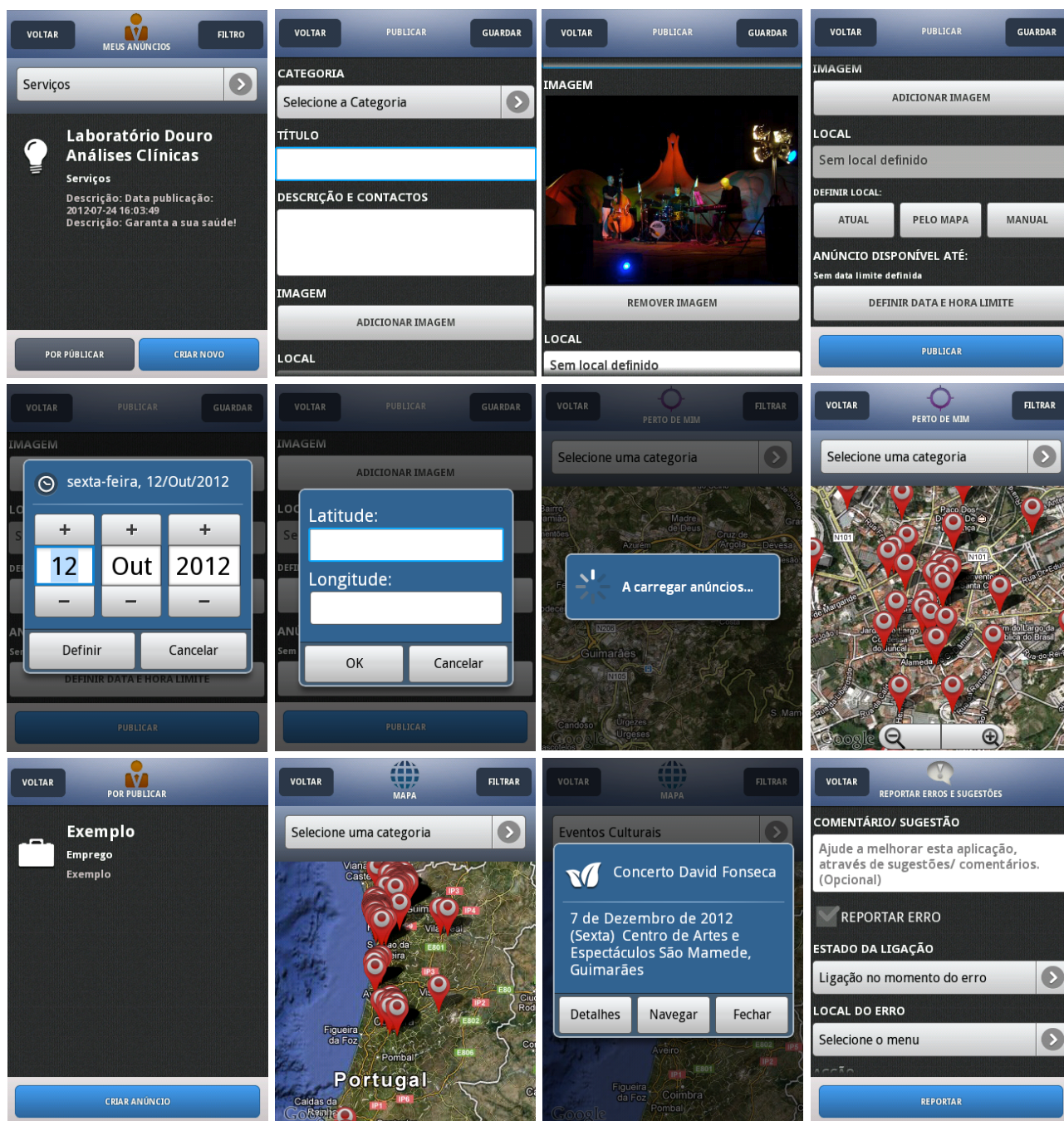


Figura D.2: Ecrãs da Aplicação (2)

Apêndice E

Web Site



GeoANÚNCIOS

DIVERSÃO NOTURNA COMPRA E VENDA EMPREGO

EVENTOS CULTURAIS SERVIÇOS RESTAURANTES

COMÉRCIO ALOJAMENTO PROMOÇÕES

Já disponível para instalação no:



Descrição: A aplicação GeoAnúncios pretende ser uma forma de facilitar a divulgação e a procura de anúncios de uma forma rápida, simples e gratuita. Englobando várias categorias, como a oferta e procura de emprego, de habitação e de serviços, diversão noturna e eventos culturais, comércio, restaurantes, e compra e venda de artigos. É por isso uma aplicação abrangente e que reúne num único local vários interesses. Sendo uma aplicação para dispositivos móveis permite que toda a informação esteja sempre disponível aos utilizadores para consulta ou publicação de anúncios. Através da georeferenciação dos anúncios, é fácil encontrar o local do anúncio e visualizar os que estão perto da sua posição geográfica.





[Download e instalação manual. Aqui.](#)

Compatível com Android 2.2 ou superior.

Figura E.1: <http://geoanuncios.dsi.uminho.pt>

Apêndice F

Divulgação

GEOANÚNCIOS

Anúncios em todo o país! Veja já em Aveiro, Braga, Coimbra, Guimarães, Lisboa e Porto.
Publique também os seus!
Tudo gratuitamente!
Aplicação desenvolvida na Universidade do Minho

Discotecas, bares e festas!
Fique a par dos locais exatos de diversão noturna e seja notificado de festas. Divulgue e partilhe também!

Esteja a par dos eventos culturais e concertos por todo o país!
Adicione também aos seus favoritos, e navegue para o local diretamente através da aplicação.

Compre ou venda artigos.

Procure emprego e entre em contacto através da aplicação com o anunciante.
Tem um cargo para oferecer? Anuncie aqui!

Conheça e divulgue serviços.

Conheça restaurantes perto de si, ou divulgue o seu restaurante ou um restaurante que aprecia.

Encontre promoções interessantes, ou divulgue as promoções da sua loja ou negócio.

Encontre ou anuncie habitações para alugar e para venda, ou quartos para estudantes.

Anúncios de grandes e pequenas superfícies comerciais.

OPÇÕES **GEOANÚNCIOS** **INFO**

NOTIFICAÇÕES (172) **VER ANÚNCIOS**

PERTO DE MIM **VER NO MAPA**

FAVORITOS **OS MEUS ANÚNCIOS**

DOWNLOAD

<http://geoanuncios.dsl.uminho.pt>

Aceda ao site através do seu Android, faça o download e instale a aplicação.
Pode aceder ao site através do QRCode, abrindo este email no Android e clicando no link ou introduzindo o endereço manualmente.

Figura F.1: Email de Divulgação – 27 de Setembro de 2012



Figura F.2: Email de Divulgação – 8 de Outubro de 2012

